Response #1

1. Have the project definitions/expectations been clearly stated?
Yes they were decently well stated. I felt that between the original project overview and the
criteria for grading we were able to figure out what needed to be done for the project.

2. Have you had access to appropriate tools for completing the projects?
Yes, the course wiki was very valuable in this. I referred many times to the documentation section
of the course wiki during almost every project to make sure that I was doing the correct procedure for
recompiling the kernel, adding modules, etc.

3. Has the level of difficulty been appropriate for this course?
I think so. I never felt that the projects were over my head, but none of them were too easy.
Without the seminars to explain the projects I would have been much more lost for the projects, but with
them it was fine.

4. Were you sufficiently prepared for the level of C programming required for these projects?
Yes, but I did just take CS 372 the term before this so it was pretty fresh in my mind.

5. Have the projects helped you to understand operating systems concepts?
Definitely, although I think it enforced my understanding of the algorithms behind them a little
more than what the algorithms were actually doing.

6. Is Linux an appropriate basis for these types of projects, or would you prefer using a
proprietary model (e.g., Windows) or a small “academic” model (e.g., Minix)?
I think Linux was good because it is something that most people have worked on and is decently
easy to work with/find help for on the internet if you are lost.

7. Have the projects helped you to understand the Linux kernel?
Yes, I have a much greater understanding of the real brains behind an operating system after taking
this class.

8. Have the projects increased your interest in career work in Linux kernel development or other
open-source projects?
Maybe a little, I like Linux but being an ECE major I think I'll be working more on the
hardware/software edge than purely on the Linux kernel.

9. Do you prefer projects in several different areas of the kernel or would you prefer projects that
go into depth on a single piece of the kernel (e.g., the I/O system)?
I liked the fact that each project focused on a different part of the kernel. It really made my
understand of each part get a lot deeper than I think it would have been had they been more general.

10. Have you asked the TAs for help on any of the projects? Have they been helpful?
Except for posting to the listserv we didn't ask the TA's for any help.

11. Did you attend the project seminars? Were they helpful? Any suggestions for additional
seminars and/or improvements to existing seminars?
I think they were perfect. They increased my understanding enough that I didn't feel lost going in to
the projects, but were broad enough that I had the ability to go in and make my own mistakes while doing
the project.

12. Have the mailing list and the course wiki been useful in understanding and completing the
projects?
Definitely, the listserv was great for posting a question to at the end of a long period spent working
on a problem. Pretty much every time we asked a question when we came back to work on it our
question was answered or we were sent on a path that lead us to an answer.

13. How much time have you been spending on the projects? Have you been able to finish the
projects?
Probably an average of 10-15 hours on each project between conceptualizing and coding. We have
finished all of the projects as well.

14. Have the projects been graded fairly?
I think so. We have gotten almost 100% on every project so it feels like it to me.

15. Other comments/suggestions:
I almost felt like I was given too much starter code on some of the projects, it ended up making the
projects easier, which was good I guess since our average time spent on a project was around 12
hours, but I didn't feel like I was writing much original code.
Response #2

1. Have the project definitions/expectations been clearly stated?
   Yes

2. Have you had access to appropriate tools for completing the projects?
   Yes

3. Has the level of difficulty been appropriate for this course?
   Yes

4. Were you sufficiently prepared for the level of C programming required for these projects?
   Yes (It’s easier to copy surrounding syntax for me than code into a blank file)

5. Have the projects helped you to understand operating systems concepts?
   Yes

6. Is Linux an appropriate basis for these types of projects, or would you prefer using a proprietary model (e.g., Windows) or a small “academic” model (e.g., Minix)?
   Linux is the best choice, I appreciated the ‘fake’ VM for the scheduling assignment though. It made testing so much easier

7. Have the projects helped you to understand the Linux kernel?
   Yes, very much

8. Have the projects increased your interest in career work in Linux kernel development or other open-source projects?
   No, I am not a CS major

9. Do you prefer projects in several different areas of the kernel or would you prefer projects that go into depth on a single piece of the kernel (e.g., the I/O system)?
   I do not really care for this class, I’m not even a CS major. I like the breadth of the class, too deep and I would really be bored.

10. Have you asked the TAs for help on any of the projects? Have they been helpful?
    Haven’t asked for help.

11. Did you attend the project seminars? Were they helpful? Any suggestions for additional seminars and/or improvements to existing seminars?
    I attended the seminars, but would’ve appreciated their slides being posted. They go kind of fast sometimes.

12. Have the mailing list and the course wiki been useful in understanding and completing the projects?
    I have a phenomenal amount of junk in my email from this mailing list I have to delete. I think using Blackboard to communicate would be much better than constantly spamming my email. Haven’t used the wiki.

13. How much time have you been spending on the projects? Have you been able to finish the projects?
    I finished the projects, spent about 6-12 hours on each one (even the first one).

14. Have the projects been graded fairly?
    I would’ve appreciated feedback on why I lost points on the evaluations. I still don’t know what I’m missing.

15. Other comments/suggestions:
    Class is good overall.

Response #3

1. Have the project definitions/expectations been clearly stated?
   Generally yes, but for project1, a lot of confusions came from unclear/ not updated information from last class. They were just minor typos or old links(such as nfs/.../fall2009), but when you are not familiar with these stuffs at all in the earlier time, it was quite hard to figure out.

2. Have you had access to appropriate tools for completing the projects?
   Yes, there were a lot of tools and sources we could make use of from the class website or from www.

3. Has the level of difficulty been appropriate for this course?
They were pretty hard even with 4 members, but in fact, figuring out what we were supposed to do was very clear and easier to understand than actually dealing with it to make it work. Level of difficulty was appropriate but it just took a lot of time generally.

4. Were you sufficiently prepared for the level of C programming required for these projects?
   Yes, but C programming level was not such a big concern for kernel coding.

5. Have the projects helped you to understand operating systems concepts?
   Definitely yes. Editing kernel really help me to understand how the kernel is being operated at the moment and the ways we can improve it.

6. Is Linux an appropriate basis for these types of projects, or would you prefer using a proprietary model (e.g., Windows) or a small “academic” model (e.g., Minix)?
   Linux is an open-source OS, I think it should be the most appropriate kernel for these projects since it is the one actually being used by many people compared to Minix which is not.

7. Have the projects helped you to understand the Linux kernel?
   Yes, similarly with answer for 4, I learned a lot about the Linux kernel. Since boot sequence editing or registry editing were all we could do when using windows, I felt I was more directly working with the kernel and that helped me understand a lot about it.

8. Have the projects increased your interest in career work in Linux kernel development or other open-source projects?
   Yes, although I am interested in other things it really helped me to understand what dealing with open-source OS and kernel was about.

9. Do you prefer projects in several different areas of the kernel or would you prefer projects that go into depth on a single piece of the kernel (e.g., the I/O system)?
   I prefer in several different areas more since this was the first class for learning kernel and we can learn various aspects of the system.

10. Have you asked the TAs for help on any of the projects? Have they been helpful?
    Yes we did and they were very helpful. They even spared extra time for helping us out even after their office hour was finished.

11. Did you attend the project seminars? Were they helpful? Any suggestions for additional seminars and/or improvements to existing seminars?
    I personally did not attend any of project seminars but my teammates did. I could say they were very helpful because the seminar helped us understand the parts we were unclear.

12. Have the mailing list and the course wiki been useful in understanding and completing the projects?
    They were probably the most helpful methods of all, but I wanted to suggest to make a thread or something else, because there were way too many emails coming from this mailing list when it gets close to the project due date, and there are several emails that I did not even open.

13. How much time have you been spending on the projects? Have you been able to finish the projects?
    With 4 senior students in a group, it was often hard to find a time in common for all, so we often ran out of time. But we were able to finish it fairly well.

14. Have the projects been graded fairly?
    Yes, I didn't have much concern for it.

15. Other comments/suggestions:
    For the next class, it would be nice to have a discussion board/thread rather than using mailing list.

Response #4

- Have the project definitions/expectations been clearly stated? I would say most of the project definitions were clearly stated. The first project was very simple to follow. Some of the definitions relied on your intuition to find an answer to the question.
- Have you had access to appropriate tools for completing the projects? Yes. My group had a few books to get information from. The computer lab in KEC is a nice place to meet with your group. The mailing list I think was the most helpful for this class. People posting their problems and getting answers for everyone to see is very helpful.
• Has the level of difficulty been appropriate for this course? I think the difficulty level was appropriate for the class. Although it was a hard class, it was not impossible.
• Were you sufficiently prepared for the level of C programming required for these projects? I could have used some more experience with C programming before taking this class. It would have made the class a little easier instead of learning a language while trying to figure out how to solve a problem.
• Have the projects helped you to understand operating systems concepts? Yes. Working on the projects gave me more of an understanding of how operating systems work.
• Is Linux an appropriate basis for these types of projects, or would you prefer using a proprietary model (e.g., Windows) or a small “academic” model (e.g., Minix)? Over the past few years I have grown fond of Linux. I think Linux was appropriate for the projects.
• Have the projects helped you to understand the Linux kernel? The Linux kernel is still overwhelming to me, but the projects did break me into grasping how it works.
• Have the projects increased your interest in career work in Linux kernel development or other open-source projects? No. I do not see myself being a Linux kernel developer, but I am open-minded to work on other open source projects.
• Do you prefer projects in several different areas of the kernel or would you prefer projects that go into depth on a single piece of the kernel (e.g., the I/O system)? It would be nice to go into depth on a single piece of the kernel, but having the variety and working on different parts of the kernel worked out well.
• Have you asked the TAs for help on any of the projects? Have they been helpful? I have not personally asked a TA for help. My group members have posted to the mailing list and that has helped.
• Did you attend the project seminars? Were they helpful? Any suggestions for additional seminars and/or improvements to existing seminars? I attended the seminars. They were helpful because I like to visually see what I should be doing and having it explained to me.
• Have the mailing list and the course wiki been useful in understanding and completing the projects? Like I said above, the mailing list was helpful for the course. There was a lot of activity and everybody provided plenty of input to help out.
• How much time have you been spending on the projects? Have you been able to finish the projects? Having a good group is important when working on these projects. My group spent many hours on the projects but we were able to complete all of them on time.
• Have the projects been graded fairly? I think the projects were graded fairly.
• Other comments/suggestions:

**Response #5**

1. Have the project definitions/expectations been clearly stated?
   I feel that the project definitions/expectations have been clearly and thoroughly stated for each project.

2. Have you had access to appropriate tools for completing the projects?
   I have always had access to the appropriate tools for completing the projects in some form or another (the cross reference was down for a while but there are others on the Internet)

3. Has the level of difficulty been appropriate for this course?
   I believe the level of difficulty has been appropriate for this course, the kernel is very overwhelming to start programming on and often feels convoluted, so I feel that the level of difficulty of each project allows for a gradual and gentle introduction to the kernel.

4. Were you sufficiently prepared for the level of C programming required for these projects?
   I feel that I was sufficiently prepared for the level of C programming required, however, the kernel requires a lot of reading to understand what is going on.

5. Have the projects helped you to understand operating systems concepts?
   The projects have helped me understand the various ways in which things are accomplished within the operating system and how these methods are written, along with the various different possibilities for solving problems.
6. Is Linux an appropriate basis for these types of projects, or would you prefer using a proprietary model (e.g., Windows) or a small “academic” model (e.g., Minix)?
I fully believe that Linux is the proper environment to teach this course and prefer it over any other platform for the projects we are doing.

7. Have the projects helped you to understand the Linux kernel?
The projects have absolutely helped me understand the Linux kernel and have shown me that it is possible to transition somewhat easily from a C program to a Linux kernel programmer (this does of course require a bit of studying but in general it is easy.)

8. Have the projects increased your interest in career work in Linux kernel development or other open-source projects?
The projects have increased my interest in open source projects more than kernel development. The kernel is very important and there are people that are very good at programming for it but I don’t think it’s for me but it is definitely interesting.

9. Do you prefer projects in several different areas of the kernel or would you prefer projects that go into depth on a single piece of the kernel (e.g., the I/O system)?
I prefer the projects as they are currently, it gives a general overview of how each segment works and how they interact with each other.

10. Have you asked the TAs for help on any of the projects? Have they been helpful?
I have not personally asked for TA assistance but they have assisted on the mailing list which was more than helpful.

11. Did you attend the project seminars? Were they helpful? Any suggestions for additional seminars and/or improvements to existing seminars?
I attended every project seminar and can say without a doubt that they thoroughly aided in the completion of the projects. I would have to say that I wish written details of the seminars were available, notes or references to follow (basically things that were discussed, line numbers of important functions, etc.) that way I don’t have to try and type what is being said.

12. Have the mailing list and the course wiki been useful in understanding and completing the projects?
The mailing list has been the most helpful in completing projects and understanding certain concepts along with the community aspect and helping other students. Without the mailing list I think the projects would be significantly more difficult and even more time consuming.

13. How much time have you been spending on the projects? Have you been able to finish the projects?
I have not been logging hours for the projects but what I can say is that I have spent several hours on each project understanding them and writing code. The only project we didn’t complete was this last one; there was an issue when we tried to compile as a module that no matter what we tried the module would—without fail—segfault; the code worked if it was compiled directly into the kernel so I have no idea what was happening.

14. Have the projects been graded fairly?
I can only speak for my group and our projects but I feel that the projects have been graded and evaluated fairly.

15. Other comments/suggestions:
The projects are concise enough to not take an eternity to program but detailed enough that we actually learn what is going on (if not just for the small portion we are working on) ; I encourage you to keep the same format and materials covered for the projects.

Response #6
1. Have the project definitions/expectations been clearly stated? Yes, all projects had very clear and definitions and expectations, many even gave detail on how groups could go about implementing and starting difficult sections.

2. Have you had access to appropriate tools for completing the projects? Yes, through the class mailing list, TAs, the in class demonstrations, and various online resources there was never a time where I felt like our group was stuck with no place to turn for outside input and help.
3. Has the level of difficulty been appropriate for this course? Yes and No, for the most part the difficulty was fine, but the amount of time given and responsibilities to other classes etc. made the projects more difficult to complete.

4. Were you sufficiently prepared for the level of C programming required for these projects? Yes, there was nothing in the C programming requirements that I had not already been introduced to, or that someone in the group had been and could explain thoroughly enough for the rest of the group to grasp for the projects.

5. Have the projects helped you to understand operating systems concepts? Yes, the project did a good job of explaining how operating systems handle the tasks we undertook, especially the Linux OS.

6. Is Linux an appropriate basis for these types of projects, or would you prefer using a proprietary model (e.g., Windows) or a small “academic” model (e.g., Minix)? I was able to understand and learn how to use/program within the Linux OS, but it would be nice to be able to compare it to other models. If possible have a few projects centered on Linux, then around Windows, etc.

7. Have the projects helped you to understand the Linux kernel? Yes, these projects have helped a lot to expand on what I have learned and been introduced to in the lecture for this class from prior courses such as 311.

8. Have the projects increased your interest in career work in Linux kernel development or other open-source projects? Yes and No, after this course working on other open source projects would be interesting, however I would probably not seek out jobs where the core of the work would center around Linux Kernel development.

9. Do you prefer projects in several different areas of the kernel or would you prefer projects that go into depth on a single piece of the kernel (e.g., the I/O system)? I would prefer having projects that involve several different areas of the kernel rather than focusing on one narrow section of the kernel.

10. Have you asked the TAs for help on any of the projects? Have they been helpful? I did not have any direct contact with the TAs, but when our group was stuck and other members of our group contacted the TAs they were always very timely and helpful.

11. Did you attend the project seminars? Were they helpful? Any suggestions for additional seminars and/or improvements to existing seminars? Yes I attended all of the project seminars, all of which were extremely helpful. One thing I think would help is having seminars before the plan of attack was due. At least one of the seminars occurred after we had to submit the plan of attack. After the seminar I was able to recall things on our PoA that had been misunderstood by our group or could have been made clearer.

12. Have the mailing list and the course wiki been useful in understanding and completing the projects? Yes, sending questions out on the mailing list and being able to view other people questions/responses was helpful in completing many of the projects.

13. How much time have you been spending on the projects? Have you been able to finish the projects? For a couple of the projects it felt that our group had to put in more time than what should have been required for a course of this length. Our group finished all projects but some were finished after the due date.

14. Have the projects been graded fairly? Yes, the TAs were very fair in their grading policy.

15. Other comments/suggestions:

Response #7

1. Have the project definitions/expectations been clearly stated?
   Yes, from a high level. The sometimes extremely difficult part of the projects has been understanding how to convert those high level expectations into the low level part of writing and modifying the kernel code. The point is to leave the point of figuring out what actual data structures and functions need to be changed within the code to us, so no complaints here.

2. Have you had access to appropriate tools for completing the projects?
   Well, we have had access to project descriptions, rubrics, TAs, and seminars, which have all been useful. A lot of our time was spent debugging the kernel and trying to figure out what our modifications did in that environment, so I think maybe a kernel debugger tutorial or a kernel debugging lecture/presentation might have been helpful.
3. Has the level of difficulty been appropriate for this course?
Kernel coding is definitely difficult for anyone without a lot of previous exposure to it, and there is really no good way to get around that, so from that standpoint it has been appropriate. On the other hand, it is assumed that a student in a 4-credit class will spend 8 hours a week outside of class reading the textbook, studying, and doing homework to be successful in the class. I think it would be more or less impossible to spend only this amount of time working on this class, because of the time-intensiveness of the projects. So in this sense, the level of difficulty is too high.

4. Were you sufficiently prepared for the level of C programming required for these projects?
Yes, but I am a transfer student, so I cannot speak about OSU’s lower-division CS courses. At MHCC we programmed exclusively with C++ in CS161, 162, and 261, so I personally felt well prepared. 311 is a prerequisite here, and it seems like anyone not knowledgeable about C or not willing to learn it on the fly would have trouble passing it and even getting to 411.

5. Have the projects helped you to understand operating systems concepts?
Definitely. Without the projects this class would have been far too abstract and theoretical. Diving into the code, modifying it ourselves, and watching the effect of those modifications has been a good way to put the theory being taught into practice, and make it more real.

6. Is Linux an appropriate basis for these types of projects, or would you prefer using a proprietary model (e.g., Windows) or a small “academic” model (e.g., Minix)?
It seems like Windows would have very limited value, because not all of the code can be directly seen and modified. Using it introduces an undesirably high level of abstraction as to what is actually going on within the OS.

Linux makes more sense: it is all open-source, and very important in the CS world. Hacking at the kernel gives us direct real-world kernel coding experience that we can take on with us when we enter the job market.

Using something like Minix also plausible though, and I believe Tannenbaum took the time create it for good reason. Its simplicity would allow students to implement more complicated changes to the operating system. Students would be able to spend more time working on the OS concept at hand, and less time trying to decipher the specific idiosyncrasies of the Linux kernel.

7. Have the projects helped you to understand the Linux kernel?
Definitely. Even though I am fairly comfortable with C, the Linux kernel is like no other kind of C that I have ever seen. Before this class, it was complete Greek to me. Now I feel like I would have some confidence modifying the kernel if necessary.

8. Have the projects increased your interest in career work in Linux kernel development or other open-source projects?
This class has kind of made me want to avoid kernel coding as a career in the future. It's pretty frustrating to debug, and because of its general obfuscated style and lack of comments it's almost never easy to understand the code that is already there.

9. Do you prefer projects in several different areas of the kernel or would you prefer projects that go into depth on a single piece of the kernel (e.g., the I/O system)?
I think that it makes more sense to have several unrelated smaller projects, instead of one term long project or series of related projects. This class speaks more broadly than deeply on the vast topic of what goes on inside of operating systems, and the assigned work should reflect that. It seems like that it would not make much sense to cover a lot of OS topics superficially in lecture and then dive into one deeply in the projects.

10. Have you asked the TAs for help on any of the projects? Have they been helpful?
We've used Tony as a resource for at least a couple of our projects. He has been extremely helpful to us in understanding precisely what is expected from us, and for helping us get on the right track when we are stuck.

11. Did you attend the project seminars? Were they helpful? Any suggestions for additional seminars and/or improvements to existing seminars?
The seminars are extremely helpful in focusing my understanding of the material as it related to the projects, I wish we had had a seminar for each and every one.

12. Have the mailing list and the course wiki been useful in understanding and completing the projects?
I have not spent much time navigating the course wiki, but the mailing list has been somewhat helpful. It has unstuck me and my team on more than one occasion.

13. How much time have you been spending on the projects? Have you been able to finish the projects?
Me and my group have all been spending what seems like an inordinate amount of time on the projects, I would say 15 hours per project per person is about average. So far through tenacity we have finished all of them and have done well grade-wise.

14. Have the projects been graded fairly?
I feel that the projects have been graded fairly. Since a rubric is supplied for each one, we’ve known what to expect from the grader and we have rarely been surprised by our results.

15. Other comments/suggestions:
There were times when it felt like more guidance in certain respects could have saved our team many hours of frustration, but perhaps this is really difficult to avoid in a kernel coding class. In particular, we did not spend much time typing in code or writing algorithms when we did our projects, we spent most of our time configuring the kernel correctly, debugging, and attempting to figure out the code that was already present. It might have been less time-consuming but still just as educational if the projects had been set up so that they were easier to configure, and if we were given more information about what the present code was doing, and if we were given more information about how to debug the kernel.

Response #8
1. The projects were pretty clear, I only remember getting hung up on project 1 and 2.
2. We did have all the tools to complete the assignments, the first project was a good intro.
3. Some of the projects were really time consuming and difficult, but I guess that's expected for a 400 level class.
4. I didn't have any trouble with the C programming, so I was sufficiently prepared.
5. The projects did help me understand the concepts of what the duties of an operating system has to do.
6. I think Linux is the best for this class.
7. The projects have definitely helped me understand the Linux kernel. Even the coding style and how to submit patches.
8. I am currently not interested in open source development, so I would have to say no.
9. I think going in to the different systems is better for a broader understanding, I think it's more helpful.
10. I have asked Tony many things, many times, and he has been extremely helpful.
11. I did attend the project seminars and they were very helpful, no suggestions here.
12. The mailing list was helpful sometimes, I used the wiki once I think.
13. The projects probably took us around 8 hours each, and we completed every project.
14. The projects were graded fairly, we had our projects working and we received good grades on every one so no complaints here.
15. Keep up the good work, it is a very time consuming class.

Response #9
1. Have the project definitions/expectations been clearly stated?
Yes, I think everything has been well written and explained.
2. Have you had access to appropriate tools for completing the projects?
Yes, I think the class emails have been the most help though, if there was anything I couldn't figure out, someone else in the class already had.
3. Has the level of difficulty been appropriate for this course?
I think the difficulty of the projects themselves have been appropriate, however we spent way too much time debugging to get the kernel to compile, which didn't really do anything to improve our understanding of the subjects of the projects (schedulers, etc).

4. Were you sufficiently prepared for the level of C programming required for these projects? Even though the kernel coding style was a little different, I think my C experience in my other classes was sufficient.

5. Have the projects helped you to understand operating systems concepts? Yes, however, as I said earlier: too much time was wasted on debugging.

6. Is Linux an appropriate basis for these types of projects, or would you prefer using a proprietary model (e.g., Windows) or a small “academic” model (e.g., Minix)? The amount of documentation available for Linux makes it ideal, it probably would have taken a lot more time to learn how the Windows OS works before being able to do any of the projects.

7. Have the projects helped you to understand the Linux kernel? Yes, CS 311 helped us understand the basics, and these projects helped with some of the more in-depth stuff.

8. Have the projects increased your interest in career work in Linux kernel development or other open-source projects? I am not a fan of Linux kernel style coding, and that has probably resulted in me not wanting to work on it again. The other open source projects I’ve worked on haven’t been as bad as the Linux kernel, so I think.

9. Do you prefer projects in several different areas of the kernel or would you prefer projects that go into depth on a single piece of the kernel (e.g., the I/O system)? More projects that don’t go as in depth that cover various parts of the kernel is more beneficial to me personally than going way in depth on one part. I think going further in depth is better taught in lecture.

10. Have you asked the TAs for help on any of the projects? Have they been helpful? Yes, and yes, they were very helpful.

11. Did you attend the project seminars? Were they helpful? Any suggestions for additional seminars and/or improvements to existing seminars? I attended all of the project seminars, and I think they were the most helpful, as far as helping me understand the project.

12. Have the mailing list and the course wiki been useful in understanding and completing the projects? There were a few places in the wiki that were out of date or incorrect, but the mailing list was one of the best resources during the class.

13. How much time have you been spending on the projects? Have you been able to finish the projects? As I said earlier, way too much time was spent on debugging, but we have been able to finish all of the projects.

14. Have the projects been graded fairly? I think everything has been graded fairly, yes.

15. Other comments/suggestions:

Response #10
1. Have the project definitions/expectations been clearly stated?
For the most part yes. They are well organized which makes it easy find the information about what is needed for each part etc, however, they are also a little verbose at times which sometimes makes things look more complicated than they really are, but at the end of the day this is probably the lesser of two evils. There were some instances where in the longer portion of the project description (like project 4 where there is the long list of functions) there seems to be “requirements” which are not tested, this could be seen as a disparity between the project documentation and the evaluation criteria which often confuses students.

2. Have you had access to appropriate tools for completing the projects?
   Absolutely, the most valuable resource in all of these projects is people. The only tools really required are the VM, SVN, SSH, and a text editor (I like notepad++). The mailing list and TAs are vital to helping students get going on these projects.

3. Has the level of difficulty been appropriate for this course?
   I think so, as an electrical engineer I don't have much in the way of programming experience compared with the rest of the class and I didn't find it so hard that I could not keep up. The only problem I ran into was that often seniors who are in senior design don't have the time for long drawn out projects that this class relies on, and this can create tension in groups especially during spring term.

4. Were you sufficiently prepared for the level of C programming required for these projects?
   Absolutely, CS311 requires as much of an understanding of C as CS411.

5. Have the projects helped you to understand operating systems concepts?
   Partially, from the standpoint of a specific algorithm and a specific implementation. The general understanding came mostly from the lectures.

6. Is Linux an appropriate basis for these types of projects, or would you prefer using a proprietary model (e.g., Windows) or a small “academic” model (e.g., Minix)?
   I think using a “real world” operating system for these classes has a lot of benefits, and either linux or windows are appropriate for this. Having that kind of experience on a resume is a good thing. I can see good arguments for using windows and linux to teach this class, windows is far more commonplace, but linux is open source and allows for full visibility of the source code.

7. Have the projects helped you to understand the Linux kernel?
   Yes

8. Have the projects increased your interest in career work in Linux kernel development or other open-source projects?
   Not really, I'm an electrical engineer so it's not a fair question.

9. Do you prefer projects in several different areas of the kernel or would you prefer projects that go into depth on a single piece of the kernel (e.g., the I/O system)?
   I prefer broader projects that allow for a good generalist understanding of the kernel, the deep understanding usually comes from on-the-job experience anyway.

10. Have you asked the TAs for help on any of the projects? Have they been helpful?
   *

11. Did you attend the project seminars? Were they helpful? Any suggestions for additional seminars and/or improvements to existing seminars?
   I attended as many as I could, and they were very helpful in clarifying the projects and what is needed to complete them.

12. Have the mailing list and the course wiki been useful in understanding and completing the projects?
   At times yes.

13. How much time have you been spending on the projects? Have you been able to finish the projects?
    I didn't finish project 3 but the rest I did, they really should only take about 5-8 solid hours (group time) each.

14. Have the projects been graded fairly?
    Yes.

15. Other comments/suggestions:
Response #1

1. Have the project definitions/expectations been clearly stated?
   Not always. There have been times when the project descriptions were vaguely worded and other times when complex steps were overly simplified.

2. Have you had access to appropriate tools for completing the projects?
   Access to the tools was always provided for the projects, but we almost always had to tweak them in some way in order to get them to work. I feel that, as students, we should be concerned about using the tools to do the project, not messing around with the tool, just to get it to function correctly before we can even begin the project.

3. Has the level of difficulty been appropriate for this course?
   For a 400 level course, I would say yes. I feel like we spent a lot of wasted time messing around with the linux kernel because of problems that had nothing to do with the projects. For the most part, though, this was pretty much as hard as I expected. I did have an awesome group full of hard workers and knowledgeable people, so that made it a little more manageable.

4. Were you sufficiently prepared for the level of C programming required for these projects?
   Yes. I had no problems with the actual C programming portion of the projects. I do wish that we had more actual coding involved. In most of the projects, it felt like most of our time was spent planning for the fewer than 50 lines of code we actually had to write.

5. Have the projects helped you to understand operating systems concepts?
   Definitely. I have learned a great deal about memory management and I/O schedulers from the last two projects.

6. Is Linux an appropriate basis for these types of projects, or would you prefer using a proprietary model (e.g., Windows) or a small “academic” model (e.g., Minix)?
   I did not like using Linux for our projects because I felt like we wasted days worth of time just trying to tinker with it to get it to function. Additionally, I think the learning curve to even understand the coding inside of the kernel is way too steep to add on top of the course load we already have. I don't know if Window is a viable option or not; I just think that Linux is a poor choice.

7. Have the projects helped you to understand the Linux kernel?
   Not really. The only thing I’ve learned is that I’m not very fond of Linux. I think that their “coding standards” are laughable at best and that their near insistence on making the code as difficult to understand as possible is shameful.

8. Have the projects increased your interest in career work in Linux kernel development or other open-source projects?
   Open source projects: yes. Linux kernel development: not a chance.

9. Do you prefer projects in several different areas of the kernel or would you prefer projects that go into depth on a single piece of the kernel (e.g., the I/O system)?
   I like that the course took us through several important parts of the operating system. I think that it would be nice to know more about some of the pieces, but that maybe that should be reserved for a separate class. This is “Operating Systems”, our focus should be on the system as a whole.

10. Have you asked the TAs for help on any of the projects? Have they been helpful?
    We have emailed the mailing list and, I believe, the TA's have been pretty good about helping with problems. As a whole, I think that they have been moderately helpful.

11. Did you attend the project seminars? Were they helpful? Any suggestions for additional seminars and/or improvements to existing seminars?
    I attended all of the project seminars. I did not find them incredibly helpful, but I was glad that I went because I always picked out a small thing or two that helped me later. My one critique is that the seminars did not feel that prepared. I always felt like they just came in and talked, without any real plan or direction.

12. Have the mailing list and the course wiki been useful in understanding and completing the projects?
Not particularly. I think that most people replied to the mailing list just to get their points, whether their answers were helpful or not. I don't think I ever really used the wiki unless the project description told me to, explicitly.

13. How much time have you been spending on the projects? Have you been able to finish the projects?
   I have spent more time on the projects for this course than I have for any other class to date. If I had to guess, I would say my group spent 20 hours per project. Because we put in the time, we were able to finish every project.

14. Have the projects been graded fairly?
   Yes, the grading accurately reflected how well we satisfied the grading criteria.

15. Other comments/suggestions:
   If the course is going to continue using Linux, I would request that you have the students use a version that does not need to be tweaked, just to get it to work in the first place.

Response #12
1. Have the project definitions/expectations been clearly stated?
   Most of the time the definitions were clear and good. Project one had some issues because some of the files were not good and buggy.

2. Have you had access to appropriate tools for completing the projects?
   Yes, all the tools needed were easily accessible.

3. Has the level of difficulty been appropriate for this course?
   Yes, I feel the level of difficulty was appropriate for this course.

4. Were you sufficiently prepared for the level of C programming required for these projects?
   At first, the level of C was difficult but it became easier.

5. Have the projects helped you to understand operating systems concepts?
   Lecture and projects together helped me understand the concepts.

6. Is Linux an appropriate basis for these types of projects, or would you prefer using a proprietary model (e.g., Windows) or a small “academic” model (e.g., Minix)
   Even though Linux was kind of annoying to work with, I feel it was appropriate.

7. Have the projects helped you to understand the Linux kernel?
   I have a somewhat understanding of the Linux Kernel

8. Have the projects increased your interest in career work in Linux kernel development or other open-source projects?
   Not really, cause some of the debugging working was a pain.

9. Do you prefer projects in several different areas of the kernel or would you prefer projects that go into depth on a single piece of the kernel (e.g., the I/O system)?
   I prefer different areas of Kernel.

10. Have you asked the TAs for help on any of the projects? Have they been helpful?
    We primarily used the knowledge obtained from the class list and the help sessions on the projects given by Rob Hess.

11. Did you attend the project seminars? Were they helpful? Any suggestions for additional seminars and/or improvements to existing seminars?
    Yes, the project seminars given by Rob Hess were extremely useful.

12. Have the mailing list and the course wiki been useful in understanding and completing the projects?
    Yes, they were both extremely useful.

13. How much time have you been spending on the projects? Have you been able to finish the projects?
    On average, we spent about 10-15 Hours on the projects.

14. Have the projects been graded fairly?
    Yes the grading on the project were fair.

15. Other comments/suggestions:
    Nope
Response #13

Have the project definitions/expectations been clearly stated?
Some of it was good step by step instructions but some of them were really un clear in what to do.

Have you had access to appropriate tools for completing the projects?
Yes, all we needed are the VMs. They served their purpose.

Has the level of difficulty been appropriate for this course?
Yes, for a 400 level course the amount of work put in was appropriate.

Were you sufficiently prepared for the level of C programming required for these projects?
No, till this day I still struggle with pointers in C. Although this class strengthened my knowledge on them, I am still not as strong at them as I would like them to be. NOTE: ECE major.

Have the projects helped you to understand operating systems concepts?
Yes conceptually they make more sense after you do the project and the “plan of attack”. The plan of attack really helps us understand how are going to approach the problem before starting.

Is Linux an appropriate basis for these types of projects, or would you prefer using a proprietary model (e.g., Windows) or a small “academic” model (e.g., Minix)?
No, But there would not be any other options. I can see that open source would be great to change and implement as compared to windows or some other proprietary software, but the coding convention and the style of Linux coding did not really help for quick learning of the materials.

Have the projects helped you to understand the Linux kernel?
Yes as explained above it did but not quickly. Our group had to jump through the Linux hurdles.

Have the projects increased your interest in career work in Linux kernel development or other open-source projects?
Neutral, I can see it would be hard to implement but challenging at the same time.

Do you prefer projects in several different areas of the kernel or would you prefer projects that go into depth on a single piece of the kernel (e.g., the I/O system)?
I believe it would be useful for people that enjoy linux to learn about different areas of the kernel. But for people that don’t I can see that it could be frustrating to learn about multiple areas just to learn about the class contents.

Have you asked the TAs for help on any of the projects? Have they been helpful?
N/A

Did you attend the project seminars? Were they helpful? Any suggestions for additional seminars and/or improvements to existing seminars?
Yes, they have been helpful. I like the way they are now how the TA would talk about sections in the code.

Have the mailing list and the course wiki been useful in understanding and completing the projects?
Yes, the wiki was helpful for looking up common problems and the mailing list had its miss and hits. At times it was very helpful when you had a common problems that people all ran into but at times it was frustrating when you had an uncommon problem and no one could help.

How much time have you been spending on the projects? Have you been able to finish the projects?
An average of 15-20 on each project. Yes we have completed all projects.

Have the projects been graded fairly?
Neutral, I really don’t think that linux coding conventions should be apart of the points. We implemented the algorithms perfectly but lost little points on coding style.

Other comments/suggestions:
None, this has been an interesting class I can say that our group have learned a lot together. Definitely keep the plan of attacks they were very helpful but maybe have them due after the TA seminar so we can be pointed in the right direction before we dive in to the code ourselves.
Response #14
1. Have the project definitions/expectations been clearly stated?
   Yes, the step-by-steps were very nice and useful.
2. Have you had access to appropriate tools for completing the projects?
   Yes, flip was available. Also, remote desktop and VM were accessible from off-campus, which is a good thing.
3. Has the level of difficulty been appropriate for this course?
   Sort of. I've felt that the projects weren't as innovative as CS311 projects because most of the projects were basically changing from one method to a different one.
4. Were you sufficiently prepared for the level of C programming required for these projects?
   Yes, after CS311, the C programming in the class isn't hard.
5. Have the projects helped you to understand operating systems concepts?
   Definitely yes. They have helped with FIFO, FCFS, Best Fit, SSTF, etc.
6. Is Linux an appropriate basis for these types of projects, or would you prefer using a proprietary model (e.g., Windows) or a small “academic” model (e.g., Minix)?
   Linux is definitely THE best for these types of projects.
7. Have the projects helped you to understand the Linux kernel?
   Parts of the kernel.
8. Have the projects increased your interest in career work in Linux kernel development or other open-source projects?
   Yes, I like the Linux community and how it works.
9. Do you prefer projects in several different areas of the kernel or would you prefer projects that go into depth on a single piece of the kernel (e.g., the I/O system)?
   Different areas. I like knowing a little of everything before I go into depth on one piece.
10. Have you asked the TAs for help on any of the projects? Have they been helpful?
    Nope, the TAs have been quite hidden from view. However, it's fine because the projects weren't too difficult.
11. Did you attend the project seminars? Were they helpful? Any suggestions for additional seminars and/or improvements to existing seminars?
    Yes, they were very helpful. They gave us a very good starting point on the projects.
12. Have the mailing list and the course wiki been useful in understanding and completing the projects?
    The course wiki has been great at the beginning of the term.
13. How much time have you been spending on the projects? Have you been able to finish the projects?
    On average, about 10 hrs on each project. And yes, we've finished every project.
14. Have the projects been graded fairly?
    Yes, the project gradings have been fair. The graders are not as harsh as graders in previous courses.
15. Other comments/suggestions:
    None

Response #15
1. Have the project definitions/expectations been clearly stated?
   Yes the project definitions/expectations have been clearly stated. As a group we had no problem understanding what was expected of us upon completion of the projects.
2. Have you had access to appropriate tools for completing the projects?
   As far as having everything that was needed in order to complete the projects we had everything that we need. A problem that we noticed was how slow the VM would be when there were a lot of people trying to work on the project.
3. Has the level of difficulty been appropriate for this course?
Yes, the level of difficulty has been appropriate for this course, the projects seemed to follow the lectures nicely so understanding the projects helped a lot with the quizzes and midterms.

4. Were you sufficiently prepared for the level of C programming required for these projects?
   Yes, I have taken several classes in the past that prepared me for the level of C programming that was necessary for completing the projects.

5. Have the projects helped you to understand operating systems concepts?
   Yes the projects have helped me to understand operating system concepts. After completing the projects I felt I had a good understanding of all of the topics that we covered in lecture.

6. Is Linux an appropriate basis for these types of projects, or would you prefer using a proprietary model (e.g., Windows) or a small “academic” model (e.g., Minix)?
   I believe Linux is an appropriate basis for these types of projects because the code is all open-source as opposed to proprietary models from Windows. Also, the use of Linux can bring more interest to open-source, such that students in the class can contribute to the operating system if they feel so compelled.

7. Have the projects helped you to understand the Linux kernel?
   The projects have helped me understand some parts of the Linux kernel (the parts that we modified) as well as the coding style of the kernel. I think it would be too much for one class to completely understand the Linux kernel.

8. Have the projects increased your interest in career work in Linux kernel development or other open-source projects?
   The projects have got me interested in Linux kernel development, maybe as a hobby, but not as a career.

9. Do you prefer projects in several different areas of the kernel or would you prefer projects that go into depth on a single piece of the kernel (e.g., the I/O system)?
   I prefer projects in several different areas of the kernel so that I could gain more of an overall understanding of the kernel rather than an in depth understanding of one part.

10. Have you asked the TAs for help on any of the projects? Have they been helpful?
    I did not ask the TAs for any help but their contributions to the mailing list were helpful in some cases.

11. Did you attend the project seminars? Were they helpful? Any suggestions for additional seminars and/or improvements to existing seminars?
    I did attend the project seminars and they were somewhat helpful. It would have been nicer if their were slides to go along with the presentation so that the seminars would seem more structured.

12. Have the mailing list and the course wiki been useful in understanding and completing the projects?
    Yes, I used the mailing list on several occasions to find the answers to problems that I was having with regards to the projects.

13. How much time have you been spending on the projects? Have you been able to finish the projects?
    I would say that projects 2, 3 and 4 were the most difficult thus resulting in around 10-15 hours total spent on each of the individual projects, excluding the first. Yes, I was able to finish the projects on time.

14. Have the projects been graded fairly?
    Yes, the projects have been graded fairly.

15. Other comments/suggestions:
    Overall the class was a good experience, it wasn't so difficult that it turned me completely away from Linux kernel development, but it was hard enough such that I did not breeze through the class.

Response #16
1. Have the project definitions/expectations been clearly stated?
   The project definitions were clear, however the grading criteria was sometimes confusing as some of the criteria referred to other/past projects. It would be very helpful to have visuals included in the project definitions similar to those in the project seminars. One difficult we had was keeping track of all of the versions of the compile config files, it would have been nice for these to be posted on the site and not
1. Have you had access to appropriate tools for completing the projects?
Yes, we were able to set up our virtual machine. For project 3 it was somewhat frustrating to be only able to see a the current screen as the kernel was booting inside of the virtual machine. For debugging purposes it would be nice to see a larger screen or the entire boot process.

2. Have you had access to appropriate tools for completing the projects?
Most of the difficulty in the projects came not from figuring out the definition, or from coding, but from debugging. Therefore, giving students help in debugging would greatly decrease the amount of time spent on the projects. I learned from doing the projects, however most of the learning does not come from the hours spent debugging which tend to just be frustrating.

3. Has the level of difficulty been appropriate for this course?
For the most part yes, I programmed a lot in CS311 and other courses. Some of the C used in the kernel is not very readable but with some effort I could figure out the sections I needed. For example, sometimes bits are masked against some constant defined in another section of the source tree.

4. Were you sufficiently prepared for the level of C programming required for these projects?
I think Linux is appropriate in that it is open source while also being used everywhere in industry. It is also nice to go job searching while being able to say that I have done Linux kernel coding.

5. Have the projects helped you understand Linux kernel?
Yes, we had the opportunity to read large amounts of kernel code and plan modifications to it.

6. Is Linux an appropriate basis for these types of projects, or would you prefer using a proprietary model (e.g., Windows) or a small “academic” model (e.g., Minix)?
I would say that the projects really made me get my hands dirty in terms of working with the kernel, however I was put off by the coding style and readability level of the kernel.

7. Do you prefer projects in several different areas of the kernel or would you prefer projects that go into depth on a single piece of the kernel (e.g., the I/O system)?
I definitely prefer projects that cover different areas of the kernel. I think I would even prefer doing even more, smaller, projects to cover even more aspects of the kernel (networking, cryptography, multimedia, video, etc).

8. Have the projects been graded fairly?
Having finished all of the projects and received high marks on all of them, I say that the grading was fair, but I think my opinion could be different had we not finished all of the projects.

9. Other comments/suggestions:
I would prefer smaller projects with smaller teams. I think that there was not enough code to be written in every project.

Response #17
1. Have the project definitions/expectations been clearly stated?
Yes. Sometimes, definitions were little vague in the documentation, but those were all clarified during the lectures.

2. Have you had access to appropriate tools for completing the projects?
Yes. It would have been better if we could have an access to kelley computer for longer time, but 2AM is good enough.

3. Has the level of difficulty been appropriate for this course?
No. This course required too much time on every project. There is no clear and accurate guideline that we can follow. Many hours have been wasted without learning to figure some simple bugs.

4. Were you sufficiently prepared for the level of C programming required for these projects?
Yes/No. Some parts are harder to understand, and some parts are easy.

5. Have the projects helped you to understand operating systems concepts?
Yes and No. As I said, it helped me to understand operation system concepts, but projects also required a lot of time that is not related to operation system concepts.

6. Is Linux an appropriate basis for these types of projects, or would you prefer using a proprietary model (e.g., Windows) or a small “academic” model (e.g., Minix)?
Linux is an appropriate basis for these types of projects, because I believe there are more resource than other models.

7. Have the projects helped you to understand the Linux kernel?
Yes, many hours the projects gave me a pretty good idea of the Linux kernel.

8. Have the projects increased your interest in career work in Linux kernel development or other open-source projects?
No, It doesn’t really inspire me to work with Linux kernel.

9. Do you prefer projects in several different areas of the kernel or would you prefer projects that go into depth on a single piece of the kernel (e.g., the I/O system)?
I have no preference, but since not everyone is interested in Linux kernel, it would be better to have a variety rather than going into depth on a single piece of the kernel.

10. Have you asked the TAs for help on any of the projects? Have they been helpful?
Yes, TAs were very helpful.

11. Did you attend the project seminars? Were they helpful? Any suggestions for additional seminars and/or improvements to existing seminars?
Yes, it helped a lot, but sometimes it seemed to go into some unrelated materials.

12. Have the mailing list and the course wiki been useful in understanding and completing the projects?
Yes. However, there are too many of them. It would be better to have a forum with the threads.

13. How much time have you been spending on the projects? Have you been able to finish the projects?
I have spent about 20 hours on the project.

14. Have the projects been graded fairly?
Yes, expection and criteria were clear.

15. Other comments/suggestions:
N/A

Response #18

1. Have the project definitions/expectations been clearly stated?
A lot of the project descriptions were infuriating vague. A lot of wasted time was spent trying to figure out what exactly was required.

2. Have you had access to appropriate tools for completing the projects?
Yes, the book sometimes helped and the linux cross reference really helped. Help with any kind of debugging however was not really available.

3. Has the level of difficulty been appropriate for this course?
The memory management project was definitely too difficult. It was nearly impossible to debug and the project was very time consuming. The other projects were sometimes difficult to understand conceptually but once you understood them the coding when easier.

4. Were you sufficiently prepared for the level of C programming required for these projects?
Yes, pointers gave us some trouble but that seems to happen regardless of what is being programmed

5. Have the projects helped you to understand operating systems concepts?
The last one, about scheduling I/O helped.

6. Is Linux an appropriate basis for these types of projects, or would you prefer using a proprietary model (e.g., Windows) or a small “academic” model (e.g., Minix)?
I would prefer Linux. It is open source, it has a lot of information about it on the internet. I use Linux everyday and I learned about it and not a small model that I would probably never use again.

7. Have the projects helped you to understand the Linux kernel?
Yes, how it organized, how it came together from many different people.

8. Have the projects increased your interest in career work in Linux kernel development or other open-source projects?
No, after dealing with Kernel panics and impossible to find bugs for hours I don't really want to dig deeper.

9. Do you prefer projects in several different areas of the kernel or would you prefer projects that go into depth on a single piece of the kernel (e.g., the I/O system)?
Wider range of areas would probably be more interesting.

10. Have you asked the TAs for help on any of the projects? Have they been helpful?
They were usually only a little bit helpful. I guess they were told not give any sort of concrete answers.

11. Did you attend the project seminars? Were they helpful? Any suggestions for additional seminars and/or improvements to existing seminars?
The seminars weren't helpful. Most of the notes I took were never used. I think a written down powerpoint would help so people can look at it later when they are actually doing their project.

12. Have the mailing list and the course wiki been useful in understanding and completing the projects?
The course wiki was somewhat helpful mainly with svn help and linux links. The mailing list helped a lot as everyone was having the same problems around the same time.

13. How much time have you been spending on the projects? Have you been able to finish the projects?
The projects took a lot of time. We weren't able to finish some of the projects.

14. Have the projects been graded fairly?
Yes, we got credit for working on the projects even if they didn't work which was greatly appreciated.

15. Other comments/suggestions:

Response #19
1. Have the project definitions/expectations been clearly stated?
   Yes, each project definition was clearly defined and given to us in a timely manner so the projects could be completed on time and in the fashion intended

2. Have you had access to appropriate tools for completing the projects?
   All the tools needed to finish the projects were given to us and were easily accessibly. Programs were even accessible off campus so coding and projects could be completed at home

3. Has the level of difficulty been appropriate for this course?
   This was a challenging course and I believe the projects reflected this very well. Nothing was handed to us and we had to learn things on our own and struggle with things on our own before answers were obtained

4. Were you sufficiently prepared for the level of C programming required for these projects?
   I feel that a class or a review in C programming would have helped for this course especially since C programming is not required by students anymore. The CS 151 class I took my freshman year does
5. Have the projects helped you to understand operating systems concepts?
The projects haven’t helped me understand directly the concepts so much as the book has. Reading the textbook material really helped me get a grasp on understanding operating system concepts.

6. Is Linux an appropriate basis for these types of projects, or would you prefer using a proprietary model (e.g., Windows) or a small “academic” model (e.g., Minix)?
I feel Windows would be more appropriate because this is the majority of today’s operating systems. I feel it would be more appropriate to use what a majority of students use in the present day.

7. Have the projects helped you to understand the Linux kernel?
Understanding the Linux kernel has definitely been a task and being able to use the projects to understand it more has been very beneficial.

8. Have the projects increased your interest in career work in Linux kernel development or other open-source projects?
These projects haven’t really increased any interest but they have definitely helped peak my awareness of the open source community and how it works.

9. Do you prefer projects in several different areas of the kernel or would you prefer projects that go into depth on a single piece of the kernel (e.g., the I/O system)?
I feel a long term project over the term of one or two more in depth concepts would be more beneficial to students because then they could learn about something more that interests them. What could be done is develop a couple large projects that are specific in certain areas and let students choose a term long project instead to help focus more on a particular interest to the student.

10. Have you asked the TAs for help on any of the projects? Have they been helpful?
The TAs have been very helpful throughout the term especially responding to e-mails when they can with their insights and suggestions on the projects.

11. Did you attend the project seminars? Were they helpful? Any suggestions for additional seminars and/or improvements to existing seminars?
I attended the project seminars but they didn’t seem to be helpful because they more or less explained the bigger picture of the projects instead of going through the actual project definitions specifically. I feel that if the seminars go over the project definition specifically it would be more beneficial.

12. Have the mailing list and the course wiki been useful in understanding and completing the projects?
Course Wiki on some of the projects have been the most helpful, but the open source environment we try and simulate with the e-mail list has been the most beneficial.

13. How much time have you been spending on the projects? Have you been able to finish the projects?
Approximately 15-20 hours have gone in to each project and we have been able to finish them all in a reasonable amount of time and haven’t been struggling with deadlines at all.

14. Have the projects been graded fairly?
I feel that the projects have been graded fairly and that we having the credit distribution grade is a fair system in determining the amount of work that is being put in by each group member.

Response #20

1. Have the project definitions/expectations been clearly stated?
Yes, those files describe the project clearly and I can figure out what to do by following the instructions.

2. Have you had access to appropriate tools for completing the projects?
Yes, like VM, SSH, SVN...

3. Has the level of difficulty been appropriate for this course?
I think so, but programming is a little bit challenging if one never used pointers, like me.

4. Were you sufficiently prepared for the level of C programming required for these projects?
Not really, before the class. Yet it helps to practice my C programming.

5. Have the projects helped you to understand operating systems concepts?
Definitely, I found that knowledge is never clear enough until you implement it in practice and programming is much more than algorithm itself.
6. Is Linux an appropriate basis for these types of projects, or would you prefer using a proprietary model (e.g., Windows) or a small “academic” model (e.g., Minix)?
   Well, I guess Window is not cheap enough to be used... I know little about Minix. Anyway, Linux is an appropriate basis, I think. I never found anything improper using it.

7. Have the projects helped you to understand the Linux kernel?
   Yes... I guess I will never look at Linux kernel if not for the projects.

8. Have the projects increased your interest in career work in Linux kernel development or other open-source projects?
   No... I always dislike programming... even the project is interesting. I may work on algorithm development.

9. Do you prefer projects in several different areas of the kernel or would you prefer projects that go into depth on a single piece of the kernel (e.g., the I/O system)?
   Either is OK. Maybe projects in several different areas are better since I'd learn most kernel concepts but may not have to go very deep.

10. Have you asked the TAs for help on any of the projects? Have they been helpful?
    I asked questions through mailist, and the TA responded quickly and is very helpful.

11. Did you attend the project seminars? Were they helpful? Any suggestions for additional seminars and/or improvements to existing seminars?
    Yes. Sort of helpful. It is useful to learn some kernel debug methods.

12. Have the mailing list and the course wiki been useful in understanding and completing the projects?
    Yes, the mailing list helps a lot. I didn’t use course wiki.

13. How much time have you been spending on the projects? Have you been able to finish the projects?
    About two or three days on each project. With help of other team members, we can finish the projects every time.

14. Have the projects been graded fairly?
    Yes, we got all scores we supposed to get. We once augued for some loss of scores and got it back later:)

15. Other comments/suggestions:
    A challenging course for me, a student without Linux or any OS background. Then it’s important to have a team and try to make sure that all members work on their parts.

Response #21

1. Have the project definitions/expectations been clearly stated?
   Yes.

2. Have you had access to appropriate tools for completing the projects?
   Usually, but it would help to have handy resources for some of the data structures and conventions used in the kernel.

3. Has the level of difficulty been appropriate for this course?
   Yes.

4. Were you sufficiently prepared for the level of C programming required for these projects?
   I think so, but some of the advanced C techniques that kernel coders use have been difficult to handle.

5. Have the projects helped you to understand operating systems concepts?
   Yes.

6. Is Linux an appropriate basis for these types of projects, or would you prefer using a proprietary model (e.g., Windows) or a small “academic” model (e.g., Minix)?
   If a microkernel would be easier to navigate and understand, I would support it. The Linux kernel is very large and complicated and it’s often hard to find things in it.

7. Have the projects helped you to understand the Linux kernel?
   Yes.
8. Have the projects increased your interest in career work in Linux kernel development or other open-source projects?
   A little bit, but kernel work seems like it would take time to get acclimatized to.
9. Do you prefer projects in several different areas of the kernel or would you prefer projects that go into depth on a single piece of the kernel (e.g., the I/O system)?
   In-depth specific projects.
10. Have you asked the TAs for help on any of the projects? Have they been helpful?
    No.
11. Did you attend the project seminars? Were they helpful? Any suggestions for additional seminars and/or improvements to existing seminars?
    They helped a bit, but I would have preferred an overview of the project structure instead of a code walkthrough.
12. Have the mailing list and the course wiki been useful in understanding and completing the projects?
    The most helpful things have been other students' questions.
13. How much time have you been spending on the projects? Have you been able to finish the projects?
    We have finished most of the projects with a few days of work.
14. Have the projects been graded fairly?
    Yes.
15. Other comments/suggestions:

   **Response #22**
   1. Have the project definitions/expectations been clearly stated?
      a. Yes. In most instances, the definitions and expectations, if not clear enough in the project definition, were more than clear enough in the project grading definition.
   2. Have you had access to appropriate tools for completing the projects?
      a. Aside from the slow svn server we ditched for github, and slow compile times from the VM, yes, we had access to everything we needed.
   3. Has the level of difficulty been appropriate for this course?
      a. I would say so. I learned a lot more about how the operating system works in this class than I would have from reading the slides or skimming a book.
   4. Were you sufficiently prepared for the level of C programming required for these projects?
      a. Yes. The projects tended to not require many pointer tricks, so the amount that I knew regarding pointers was more than enough.
   5. Have the projects helped you to understand operating systems concepts?
      a. I would say so. I had a horribly over-complicated idea of how each of the separate parts of an OS work, but now that I've looked at the Linux kernel, I see that most of the implementation is elegant and simple.
   6. Is Linux an appropriate basis for these types of projects, or would you prefer using a proprietary model (e.g., Windows) or a small "academic" model (e.g., Minix)?
      a. Linux is appropriate. The open code allows for us to understand what's going on, and I can't imagine doing this level of kernel editing on Windows. With regard to Minix, I have no experience with an "academic" model, but I think using something that's a real implementation as opposed to an "academic" implementation is much more useful.
   7. Have the projects helped you to understand the Linux kernel?
      a. To some small extent. I understand a lot more about how each module is designed, and how they can be swapped out easily, but not about the areas of the kernel we did not study, though I do not know what those would be (device drivers?)
   8. Have the projects increased your interest in career work in Linux kernel development or other open-source projects?
      a. Yes. I'm actually thinking of looking at the beginning steps of kernel hacking this summer, with the kernel janitor project.
9. Do you prefer projects in several different areas of the kernel or would you prefer projects that go into depth on a single piece of the kernel (e.g., the I/O system)?
a. The smaller projects are easier to manage. I can focus on one area to learn and manipulate. If I had to learn several in two or three weeks, the projects would probably be almost impossible to complete.

10. Have you asked the TAs for help on any of the projects? Have they been helpful?
a. We did not ask a TA for help, but he did reply on the mailing list a few times and was more than helpful.

11. Did you attend the project seminars? Were they helpful? Any suggestions for additional seminars and/or improvements to existing seminars?
a. The project seminars were helpful in learning the current implementation, but not always what we were going to do. Sometimes, that was a matter of how complex the current implementation is, so there isn't much to do about that.

12. Have the mailing list and the course wiki been useful in understanding and completing the projects?
a. Yes, yes, yes, a thousand times yes. They were more than helpful in answering a few questions I had, and as a general reference for what the server addresses were.

13. How much time have you been spending on the projects? Have you been able to finish the projects?
a. We have been able to MOSTLY finish the project, with roughly 10 hours to 25 hours per person spent on the project.

14. Have the projects been graded fairly?
I would say so. Any points we lost, I would agree with.

15. Other comments/suggestions:

Response #23

1. Have the project definitions/expectations been clearly stated?
I believe that they have. The project definitions contained enough direction for our group but at the same time left enough for us to work on to make each individual project challenging.

2. Have you had access to appropriate tools for completing the projects?
Throughout working on the projects, there was always the ability to check for solutions, or methods of working toward solutions, on the internet. When that turned up nothing, I usually resorted to reading the mailing list which always contained the answers that I needed.

3. Has the level of difficulty been appropriate for this course?
As a 400 level computer science course, I believe it was of appropriate difficulty. The material was challenging enough to keep me (mostly) interested but at my current level of academic progress, nothing has been impossible in the course.

4. Were you sufficiently prepared for the level of C programming required for these projects?
A little bit more preparation would have been nice but after taking a few classes and having previous C++ experience, coding in C wasn't that difficult. I would like to mention, however, that many students that have taken this class in the past that I know have complained about not being prepared enough for the projects.

5. Have the projects helped you to understand operating systems concepts?
Yes. I'm more of a theory person than a practice so I feel that the lecture was a bit better for me but getting in some practice in actual implementations is always useful.

6. Is Linux an appropriate basis for these types of projects, or would you prefer using a proprietary model (e.g., Windows) or a small "academic" model (e.g., Minix)?
I don't think that a lot of these projects are doable on a proprietary system. A lot of the development field is using Open Source projects or is for OS projects. I, personally, do not want to be doing work on a proprietary level so I feel that using Linux is a good choice. I also personally believe that Linux, whatever your flavor, is a much better operating system than Windows. Using something like Minix might be nice because it could possibly cut down on compile times but would be unnecessary.

7. Have the projects helped you to understand the Linux kernel?
Yes. I had a basic understanding of the Linux kernel coming into this class but I feel that I have a better understanding now that the class is all but finished.

8. Have the projects increased your interest in career work in Linux kernel development or other open-source projects?
No, not especially. I have had some interest in working in open source development but I think that I'm going to tend away from development as a career choice.

9. Do you prefer projects in several different areas of the kernel or would you prefer projects that go into depth on a single piece of the kernel (e.g., the I/O system)?
As a general course on operating systems, I don't think that the class projects should go further into depth into each specific piece of the kernel. If a student wishes to delve more deeply into each aspect, it should be their own choice or an option they can pursue as using their restricted electives if courses are offered for them.

10. Have you asked the TAs for help on any of the projects? Have they been helpful?
I have not asked the TAs for help personally.

11. Did you attend the project seminars? Were they helpful? Any suggestions for additional seminars and/or improvements to existing seminars?
I felt that they were useful. I have no suggestions for this class in the future.

12. Have the mailing list and the course wiki been useful in understanding and completing the projects?
Yes. The course mailing list was incredibly helpful. Whenever there was a question that I wished to ask on the mailing list, someone had already asked it. It is also useful for getting started or if we got stumped on any of the projects.

13. How much time have you been spending on the projects? Have you been able to finish the projects?
We were able to get the projects mostly completed. The reason for the use of “mostly” is because we always started the projects relatively late. Making the mistake of slacking off and starting on the projects relatively late, I always felt that starting a day earlier would have been much wiser of a choice for the group.

14. Have the projects been graded fairly?
Yes, I believe they have been graded fairly.

15. Other comments/suggestions:
Dear Mother Teresa: Do not retire. This class being taught by one of the other instructors would increase the difficulty way too much and I feel that students would get much less out of it.

Response #24

1. Have the project definitions/expectations been clearly stated?
   a. I believe they were, in some cases more information on how to debug or even further tips on implementation would have been very useful.

2. Have you had access to appropriate tools for completing the projects?
   a. Yes, we had all the tools we needed.

3. Has the level of difficulty been appropriate for this course?
   a. Yes.

4. Were you sufficiently prepared for the level of C programming required for these projects?
   a. Yes, but some of my team members were not.

5. Have the projects helped you to understand operating systems concepts?
   a. Yes, it’s been really useful to know a lot more about the core functionality of operating systems.

6. Is Linux an appropriate basis for these types of projects, or would you prefer using a proprietary model (e.g., Windows) or a small “academic” model (e.g., Minix)?
   a. I think Linux was the best choice for this course, since the knowledge gained is applicable in the real world, and it’s open source, so it’s good to get more people involved in that.

7. Have the projects helped you to understand the Linux kernel?
a. Absolutely, I understand considerably more about the Linux kernel.

8. Have the projects increased your interest in career work in Linux kernel development or other open-source projects?
   a. While the class has not had an effect on my career choice, it’s had an effect on the projects I do in my spare time.

9. Do you prefer projects in several different areas of the kernel or would you prefer projects that go into depth on a single piece of the kernel (e.g., the I/O system)?
   a. I enjoyed having the variety in different areas of the kernel. I think that it would have been worse to just focus on one area.

10. Have you asked the TAs for help on any of the projects? Have they been helpful?
    a. We asked Tony for help multiple times, and he was usually helpful.

11. Did you attend the project seminars? Were they helpful? Any suggestions for additional seminars and/or improvements to existing seminars?
    a. The project seminars made the projects very clear. They were very useful to have, and additional seminars for all the projects would be great.

12. Have the mailing list and the course wiki been useful in understanding and completing the projects?
    a. The mailing list was really useful, whenever we ran across a problem, we checked the mailing list and often found the answer, if not, we could post a question.

13. How much time have you been spending on the projects? Have you been able to finish the projects?
    a. We spent a lot of time on these projects. More time than any other class. We haven’t been able to finish all the projects. I believe if everyone in the group was a capable coder, we wouldn’t of had issues.

14. Have the projects been graded fairly?
    a. I think they were.

15. Other comments/suggestions:

**Response #25**

1. Have the project definitions/expectations been clearly stated?
   Yes, the project definitions/expectations were very clearly stated. The pdf project descriptions were extremely detailed and helpful for completing the projects.

2. Have you had access to appropriate tools for completing the projects?
   Yes, as far as I know my group had access to any/all tools necessary for completing the projects.

3. Has the level of difficulty been appropriate for this course?
   Yes, the projects definitely had frustrating moments, and were difficult at times but this is to be expected at this level of a class.

4. Were you sufficiently prepared for the level of C programming required for these projects?
   Yes, luckily enough of my prior classes used C as the primary coding language. I’ve heard that other people have a majority of their experience in java. Not having at least some experience with pointers would make these projects a little too difficult.

5. Have the projects helped you to understand operating systems concepts?
   Absolutely, before this class I had never really thought about all of the things an operating system takes care of and what it provides to its users. I suppose since I've always had an OS to work with I've always taken what it provides for granted.

6. Is Linux an appropriate basis for these types of projects, or would you prefer using a proprietary model (e.g., Windows) or a small “academic” model (e.g., Minix)?
   Because of the openness of the linux kernel I think it is perfect for this class, although I can't help but think it would be interesting to work at least a few projects using the windows model. I don't think Minix would bring much excitement/interest to anyone.

7. Have the projects helped you to understand the Linux kernel?
   Yes, I understand the linux kernel much more than I did before. I also realize that the linux kernel is quite large and complex so as far as fully understanding the linux kernel no, it'd probably take much more time than this class has to offer.
8. Have the projects increased your interest in career work in Linux kernel development or other open-source projects?
   I suppose I feel I would be better prepared to work on any open-source projects and would not make as many n00bie mistakes.

9. Do you prefer projects in several different areas of the kernel or would you prefer projects that go into depth on a single piece of the kernel (e.g., the I/O system)?
   Its might better to focus on one part of the kernel so you have a more thorough understanding of how the kernel handles things -but this class did a pretty good job balancing the requirements of projects.

10. Have you asked the TAs for help on any of the projects? Have they been helpful? I have only asked the mailing list for help but the TAs have answered and been helpful.

11. Did you attend the project seminars? Were they helpful? Any suggestions for additional seminars and/or improvements to existing seminars?
   Yes I attended all project seminars and felt they were helpful. It might be nice if they provided a little bit more of a how it relates to which part of the project sometimes.

12. Have the mailing list and the course wiki been useful in understanding and completing the projects?
   Yes, the mailing list was a lot of fun in this class and both it and the wiki were useful and in some cases necessary to completing the projects.

13. How much time have you been spending on the projects? Have you been able to finish the projects?
   We have been spending probably around 10 hours a project and have had difficulty completing some of the projects on time -but we started late on a few.

14. Have the projects been graded fairly?
   Yes, the projects seem to be graded fairly.

15. Other comments/suggestions:
   This is a fun class overall, even with the stress of trying to finish a project by the deadline.

Response #26
1. Have the project definitions/expectations been clearly stated?
   Yes. Usually the project definition is clear enough about the goals of the projects. The project evaluation further states the expectation of the project in different aspects.

2. Have you had access to appropriate tools for completing the projects?
   Yes, I could find the link/documentation on the course wiki webpage that I need to complete the project most of the time.

3. Has the level of difficulty been appropriate for this course?
   Yes. We were a little bit lost for the first project, since we did not know whether we were on the right track. But the course wiki has had enough instructions to guide us step by step through the whole process. The later projects are about the level of difficulty as I expected.

4. Were you sufficiently prepared for the level of C programming required for these projects?
   Yes, I think so.
   I’m more familiar with C++ I must say. But since C++ and C aren’t all that different, and I have had some practice in C, the programming part wasn’t beyond my programming skills.

5. Have the projects helped you to understand operating systems concepts?
   Certainly.
   Projects helped me understand the algorithms from a different perspective than when I learned them in class. By actually implementing the algorithms, I got to see not only how the algorithms work, but also how they interact with the Linux kernel.

6. Is Linux an appropriate basis for these types of projects, or would you prefer using a proprietary model (e.g., Windows) or a small “academic” model (e.g., Minix)?
   Linux is appropriate in that it is open source, so we can dig as deep as we want when we’re interested in a particular aspect of it. But it would be nice if we could have some examples on the comparisons
of Linux and Windows because I believe some of us are more comfortable with windows. The examples about Windows would be more intuitive to some of us.

7. Have the projects helped you to understand the Linux kernel?
Yes. Each project is about a different aspect about Linux kernel (installing module, CPU scheduling, I/O scheduling, etc). I think I learned more about Linux kernel than I would’ve without projects.

8. Have the projects increased your interest in career work in Linux kernel development or other open-source projects?
Not really. As interesting as kernel programming is, I don’t think I will pursue my career in this field.

9. Do you prefer projects in several different areas of the kernel or would you prefer projects that go into depth on a single piece of the kernel (e.g., the I/O system)?
I would prefer to learn more areas about the kernel rather than go deep in one aspect.

10. Have you asked the TAs for help on any of the projects? Have they been helpful?
Yes. We asked Tony about the permission on SVN. We found his response energetic and helpful.

11. Did you attend the project seminars? Were they helpful? Any suggestions for additional seminars and/or improvements to existing seminars?
Yes. They were helpful to certain extent, but most of the time I’m listening to the stuff that I’ve already figured out on my own. I understand that a seminar can’t possibly cover every one’s questions, but I think it might be more helpful if instead of seminar, we have some explanation on the course wiki about the source code, and people can add their own understanding to it. In this way, we could just directly go to the part that we have problem with rather than waste time going through the whole thing only to find what confused us remains confusing.

12. Have the mailing list and the course wiki been useful in understanding and completing the projects?
Yes. People on the mailing list sometimes run into the same problem as we do. So when we are having a headache trying to figure out what went wrong, we could sometimes find the answer posted on the mailing list. Sometimes we could see posts about a particular mistake that people tend to make, so that we can learn from their mistake and get around it. The mailing list has been really helpful.

13. How much time have you been spending on the projects? Have you been able to finish the projects?
On each project, I spend about 30 hours doing research on what we need to do and doing it. We have been able to finish all the projects in time.

14. Have the projects been graded fairly?
Yes. In the project evaluation, each aspect of the project is graded, which makes it really clear what we did well in and what we need to improve. I quite appreciate it.

15. Other comments/suggestions:

Response #27

1. Have the project definitions/expectations been clearly stated?
Sometimes the culmination of things that need to be turned in with the .tar can be kind of vague but other than that the directions and expectations have been clear.

2. Have you had access to appropriate tools for completing the projects?
Yes, the virtual machines were well implemented and easy to work with. The first project seemed like it was a little too much work getting the correct Linux version installed.

3. Has the level of difficulty been appropriate for this course?
The difficulty of the second project was about right but after that it was nearly always the same thing that we were doing in the other projects which took away from the challenge.

4. Were you sufficiently prepared for the level of C programming required for these projects?
There were a few things that I needed to touch up on in order to be very successful. Mainly the level of use of pointers threw us off at the beginning when we were trying to remember what exactly had to be done to get pointers functional.

5. Have the projects helped you to understand operating systems concepts?
The plans of attacks were a good idea in trying to get us to understand what exactly the code was supposed to do. However, after project 2 it was all pretty much the same… we have a basic
implementation of a component and change the algorithm a bit... it was a bit repetitive which took away from what I felt I should be learning.

6. Is Linux an appropriate basis for these types of projects, or would you prefer using a proprietary model (e.g., Windows) or a small “academic” model (e.g., Minix)?
   I think that Linux is a good choice because it is a very popular open source alternative where changing code is easy and encouraged. It is also a good introduction for people who want to get into making changes to a large open source project.

7. Have the projects helped you to understand the Linux kernel?
   As far as the Linux kernel itself goes I feel that what I really learned that was important had to do with creating system calls and how to deal with the built in list data structure more so then how the kernel does specific things like cpu management and memory management.

8. Have the projects increased your interest in career work in Linux kernel development or other open-source projects?
   Not so much. The projects got a little redundant after the second project and didn’t do much to further increase any interest I had in working with open source.

9. Do you prefer projects in several different areas of the kernel or would you prefer projects that go into depth on a single piece of the kernel (e.g., the I/O system)?
   Maybe more of a combination of both... like have 2 projects on 1 system and 2 on another... the way it currently is, it feels like we are just altering a few algorithms to change something a small bit.

10. Have you asked the TAs for help on any of the projects? Have they been helpful?
    We never specifically asked a TA something but we have used the course community email which was fairly helpful some of the time.

11. Did you attend the project seminars? Were they helpful? Any suggestions for additional seminars and/or improvements to existing seminars?
    I think the seminars were a very good resource in trying to get these projects done. I also like how we had to do the plan of attacks first before we were directly shown how the code worked. The first seminar was a little vague though I think because it was a larger code base that this was the problem but it didn’t really feel like it was as well ordered as the others.

12. Have the mailing list and the course wiki been useful in understanding and completing the projects?
    The wiki helped a lot when trying to get stuff with the kernel to work... not necessarily just information dealing with code; I feel the mailing list was more helpful as far as coming up with actual code goes.

13. How much time have you been spending on the projects? Have you been able to finish the projects?
    On average the number of hours we have been spending on projects is around 10 hours over the span of a few days. The first project was very rushed I felt cause we only had the weekend to do it and there were a lot of things that can go wrong that can tie up a group for a few hours.

14. Have the projects been graded fairly?
    I have had no problems with how the projects have been graded.

15. Other comments/suggestions:
    Overall I think that the projects need a bit more variety since all of them felt like we were just altering an algorithm and that was it... it didn’t really feel like we were working with a massive system that has a lot of different things going on.

Response #28
1. The project requirements were clear. On the first project that we needed to turn in a plan of attack, however, it was stated in the definition that we were only supposed to talk about implementing one aspect of the project, then we lost points for not talking about the rest of the project, so that could be cleared up a bit.
2. The vm's were easy to access but really slow and sluggish.
3. The level of difficulty varied. The memory scheduler project was extremely hard, while the i/o scheduler was simplistic. Making the projects get progressively harder would have been a better plan than having the difficulties of each subsequent project change drastically.
4. The C i know was sufficient. The only trouble i had (as with most i assume) was with
pointers, which just need practice.
5. I understand the parts of the operating system that we tore apart and rebuilt, but nothing more.
6. I liked using the Linux kernel because i use it on my own machines, so i can go change my own OS if i want.
7. With the projects i understand some of the reasons linux does certain things, and how they get done.
8. I want to learn a little more about the kernel and see if i can find a way to contribute, but this has sparked a new interest in Linux for me.
9. I liked the way we did it, with a different project in a different part of the kernel.
10. I did not ask the TA's for help.
11. We didn't have any seminars
12. The mailing list has been a great tool for sharing knowledge between classmates. Other than the introductory tutorials, however, i didn't use the wiki.
13. We were able to finish all the projects. It usually took all 4 of us one or two full nights to accomplish, so about 16 hours per person.
14. The projects were graded as the grading requirements stated, so yes.

Response #29
1. Have the project definitions/expectations been clearly stated?
The project definitions/expectations provided a good overview for the end goal of the project. Implement sstf , for example. What the project descriptions didn't include were details on how to accomplish the end goal. For some of the things (such as setting up the virtual machine), there were tutorials which helped a lot. However for the projects without tutorials or specifics about the functions, it was sometimes hard to figure out what needed to be done in each function. I don't think giving this information away would be a good idea, since figuring it out is what the project is about, but more of a description on how those functions are called and used would be useful. Each time we had a function write, we came up with something that eventually worked, but we were never sure exactly what would be called by what. I think I learned the most on the second project, when we had the cpu to test the code on because we were able to look through and see where the function was called and when.

2. Have you had access to appropriate tools for completing the projects?
Yes. We had some issues running out of space while checking out our projects with all the branches we had to make. Eventually we moved everything to the vm, which helped, but since we needed a log from the svn commits, we had to set up users on the vm and then each of us had to check out a copy. This meant we could only check out one branch at a time, and would have to delete old branches to keep from running out of space. I asked on the mailing list what other groups were doing, and most said they just checked it out on there laptops. Perhaps a solution could be to allow us to just have one user on the vm with the all the branches checked out for the team, then when we do a commit, just add our name to the message?

3. Has the level of difficulty been appropriate for this course?
Yes. The projects were challenging, but we learned a lot completing them and figuring them out.

4. Were you sufficiently prepared for the level of C programming required for these projects?
Almost. Some of the coding standards were a little difficult to figure out at first. Such as passing the struct's, and the linux linked list.

5. Have the projects helped you to understand operating systems concepts?
Yes. Each of the projects helped us understand how a different part of the OS is handled. Memory, I/O, processes. Before the course I really had no clue how any of these things were actually handled. I understood that the OS decides what to schedule when when it comes to each of those things, but I didn't know what was going on under the hood.

6. Is Linux an appropriate basis for these types of projects, or would you prefer using a proprietary model (e.g., Windows) or a small “academic” model (e.g., Minix)?
I liked using Linux. Its nice to be able to tell someone that I wrote an I/O scheduler for Linux. The benefit of using a smaller system, would be that we (in theory) would understand everything about what we were doing and debugging might be easier, however I don't think these outweigh the benefits of actually working with Linux. In addition to the projects, I also learned how to compile the Linux kernel and how to open source stuff works while working on a project that we could immediately see what we were doing.

7. Have the projects helped you to understand the Linux kernel?
Yes. Its not quite so intimidating to look through all the files in the kernel. I also know now how to compile the kernel and boot the images I created.

8. Have the projects increased your interest in career work in Linux kernel development or other open-source projects?
Yes. The idea of open source was a little intimidating to me at first, because I really knew nothing about it. I've used Linux, but never to its full potential and now that I know more about how it works, I am much more interested in using on my desktop, and possibly in my future career.

9. Do you prefer projects in several different areas of the kernel or would you prefer projects that go into depth on a single piece of the kernel (e.g., the I/O system)?
The different areas for the projects helped us learn more about how different parts work, but going into dept on one of these would mean we would learn a lot about one part. For me, I like knowing in detail how everything I work is being used, but since writing an OS from scratch isn't possible in 10 weeks, I think giving a brief overview of different problems is very beneficial.

10. Have you asked the TAs for help on any of the projects? Have they been helpful?
The mailing list seemed to work best for getting questions answered quickly. This sometimes meant that the TAs did answer the question, and sometimes it didn't. We didn't actually have that many questions that someone in the group didn't know or we couldn't figure out by reading the book or discussing amongst ourselves. Therefore it was hard to get all the contributions on the mailing list.

11. Did you attend the project seminars? Were they helpful? Any suggestions for additional seminars and/or improvements to existing seminars?
The project seminars were great. The only draw back was that these were done usually after the plan was due, which maybe is best because we had to work a little harder to figure out what was going on before the seminar, but it also meant that we weren't sure what to put into the plan sometimes. We put down what we thought we needed to do, but we weren't sure. So we were usually worried the plans wouldn't be good enough.

12. Have the mailing list and the course wiki been useful in understanding and completing the projects?
The course wiki helped a lot when we needed the files in the documentation section. The mailing list however was a little intimidating. I answered a question for the first project and got a slightly rude response from someone about what I should be done rather than what I suggested they do. This doesn't seem like the spirit of open source. Or perhaps it is, either way I was hesitant to put anything else on the mailing list after that.

13. How much time have you been spending on the projects? Have you been able to finish the projects?
We spent between 10 – 20 hours on each project. The only problem with the timing, is that it was hard to find a time when four people could meet, and it wasn't easy to split these projects into sub sections.
This meant that sometimes the code would be written in one 8 hour sitting, and if one person couldn't make it, there wasn't much left for them to do. It wasn't fair in that case to give them less credit, because it could have been any of us who couldn't make it to that meeting time. Perhaps offering a few points extra credit for each project would allow people to say, O.K. this person did 30%, without the person who couldn't make it taking a hit to there grade.

14. Have the projects been graded fairly?
Yes. We didn't have any problems with the grade it seemed fair and correct.

15. Other comments/suggestions:
   Excellent class!
Response #30

1. Have the project definitions/expectations been clearly stated?
The project definitions have been complete and concise.

2. Have you had access to appropriate tools for completing the projects?
While the tools have been accessible, I believe that putting more emphasis on them would have been beneficial. Specifically, I believe it was Rob Hess who mentioned using the kernel debugger (kgdb) to help diagnose problems within your kernel code. An excellent suggestion if you have previous experience with the tool; however, if you fail to fall into that category, there's little time outside of class to try and blindly learn it for use in the projects. The only options left to you are using a barrage of printk and BUG_ON macros.
The point is I believe it would be useful to devote even a partial lecture to the basics of using the kernel debugger. I know it would have alleviated a lot of hassle from our group.

3. Has the level of difficulty been appropriate for this course?
Considering the array of topics we're required to cover, I would say the difficulty is sufficient. The project's themselves aren't terribly difficult, but all the research that has to go into figuring out how to shoehorn the relatively simple code into the workings of the Linux kernel does take up a fair portion of time.

4. Were you sufficiently prepared for the level of C programming required for these projects?
I was, but the rest of my group was left asking me questions regarding some of the "ugly pointer/macro tricks" that were occasionally used inside the kernel. With most of the CS courses moving to Java, I'm not sure there's a sound solution to alleviate this problem however.

5. Have the projects helped you to understand operating systems concepts?
Honestly, I'm not sure. The projects were beneficial to learning how Linux works. Additionally, it was beneficial to get to work through a "real" existing code base, which is something that I haven't experienced in the other CS classes (toy code provided by instructors doesn't count). I can't however say that the projects aided in my understanding of the operating system concepts. It was more a way to see how those concepts were realized inside an actual operating system.

6. Is Linux an appropriate basis for these types of projects, or would you prefer using a proprietary model (e.g., Windows) or a small "academic" model (e.g., Minix)?
Whether to adopt a Windows or Linux platform is largely irrelevant. The platform was a means to realize the concepts inside a real operating system environment. Therefore, whether Windows or Linux is merely a means of pleasing one group or another.
Now when Windows/Linux is put against a purely academic system like Minix, I believe working with the real operating systems is far more beneficial.

7. Have the projects helped you to understand the Linux kernel?
Again, as I mentioned above I believe this is the true strong points of the projects. Do I plan on becoming a kernel hacker in my lifetime, not likely. However the knowledge gained from performing the projects will carry over into other projects without question.

8. Have the projects increased your interest in career work in Linux kernel development or other opensource projects?
Sadly no, but I admit I'm what some consider to be a "Code Nazi." Most (yes, this is a generalization) of the code conventions used inside open source projects are strongly against my practices, and these feelings go beyond aesthetic nuance. Code readability for projects that hope to receive any kind of maintenance is critical, and several areas of the Linux coding conventions are adamantly against this mentality.
The projects have increased my desire to seek out other open source projects however.

9. Do you prefer projects in several different areas of the kernel or would you prefer projects that go into depth on a single piece of the kernel (e.g., the I/O system)?
The answer to this is a toss-up. Ideally, in-depth projects would be better. It would remove the vast majority of setup time involved in just figuring out the HOW of the projects. More time would be spent writing and, understanding the code, rather than trying to decipher how the Linux subsystems interconnect. The problem is of course what area do you choose? It's not as though one subsystem is somehow "more important" than another.
I do believe that the projects that had the plan of attack were the most successful. Now only were you forced to think ahead before jumping into trying to code the project, but the additional week to devise a strategy was also a useful addition to the projects.

10. Have you asked the TAs for help on any of the projects? Have they been helpful?
Our group didn't utilize the TA's outside the mailing list. I'm what most would call old-fashioned; I prefer to figure out problems on my own--or not at all.
11. Did you attend the project seminars? Were they helpful? Any suggestions for additional seminars and/or improvements to existing seminars?
I attended all the seminars, and they are CRUCIAL to complete the projects in a timely manner. Unless you have previous kernel development experience with Linux, most of the time is spent understanding how the code works together. Having someone go through the code, and ask questions of when the case arises was a real blessing.

12. Have the mailing list and the course wiki been useful in understanding and completing the projects?
There were times that the mailing list was helpful, but I have to say, it was largely useless. The people who truly wanted to be helpful were helpful. The other people merely talked to get their points. Having a grade attached to contribution only guarantees the quality of the questions/comments will be subpar. On numerous occasions, people wouldn't even read through the entire question before answering it, or fail to read earlier messages and just state the same thing over again.
The wikis and additional course documentation were very useful however.

13. How much time have you been spending on the projects? Have you been able to finish the projects?
Our group met almost every Monday, Wednesday, Friday for at least 5 hours a session. Additionally, some nights were spent ordering pizza and staying until 10PM or midnight. The majority of situations like this were due to debugging issues, which is why I mention it in the tools section above.
With the exception of the IO scheduler, all the projects were completed. In the case of the IO scheduler, the fault was with our algorithm implementation. Had we started with the correct one from the start finishing it would have been trivial as well.

14. Have the projects been graded fairly?
Yes. My only "complaint" with project grading is in the time it took to get some of the project grades. Because grades are compartmentalized, we were unable to decide whether to perform the extra credit or not because of not knowing the scores. I realize it would have been nice to just do the extra credit just in case, but free-time isn't something our group had a plethora of.

15. Other comments/suggestions:
Nothing that has not been already stated comes to mind

Response #31

1. Have the project definitions/expectations been clearly stated?
The structure of the class was very good, it was very easy to understand what was required. The projects were very interesting but the projects themselves are much harder to understand. The projects were very well written but some instruction such as SSH were harder to do unless we knew any prior knowledge. I think that if the project grading criteria was posted earlier so that the complexity would be reduced instead of guessing what would be graded.

2. Have you had access to appropriate tools for completing the projects?
Yes. We had some issues running out of space while checking out our projects with all the branches we had to make. Eventually we moved everything to the virtual machine, which helped with space issues. This meant we could only check out one branch at a time, and would have to delete old branches to keep from running out of space. We ended up taring all the files to reduce space but providing more space in the future would be a better choice.

3. Has the level of difficulty been appropriate for this course?
Yes. The projects were challenging, but I learned a lot completing them. Since my team did a lot of research before we began coding which helped out a lot.

4. Were you sufficiently prepared for the level of C programming required for these projects?
Almost. Some of the coding standards were a little difficult to figure out at first. Such as passing the struct's, and the linux linked list.

5. Have the projects helped you to understand operating systems concepts?
Yes. Each of the projects helped me understand how different parts of the OS is handled. I transferred from PSU before I came to OSU so I have about 8 years of C and C++ experience.

6. Is Linux an appropriate basis for these types of projects, or would you prefer using a proprietary model (e.g., Windows) or a small “academic” model (e.g., Minix)?
I liked using Linux. Its nice to be able to tell someone that I wrote kernel code for Linux. The benefit of using a smaller system, would make understand the basics better but using a relive system was a better choice for my learning experience and also for resume purposes. In addition to the projects, I also learned how to compile the Linux kernel and how to open source stuff works.
7. Have the projects helped you to understand the Linux kernel?
Yes. Before I took this class I didn't know how to add code or compile it so it is very good knowledge I know.

8. Have the projects increased your interest in career work in Linux kernel development or other open-source projects?
Yes. I have always been interested in open source and wanted to always work for red hat or another open source company. The reason why I transferred to OSU was because of this class and the open source community on campus.

9. Do you prefer projects in several different areas of the kernel or would you prefer projects that go into depth on a single piece of the kernel (e.g., the I/O system)?
The different areas for the projects helped me learn more about how different parts worked, but going into dept on one of these would allow a deeper understanding of a particular part but by learning different parts of the kernel, it allowed me to get a better general understanding as a whole.

10. Have you asked the TAs for help on any of the projects? Have they been helpful?
I never spoke to a TA in person but the mailing list was extremely helpful especially since it was so fast at getting back responses. So requiring the use of the mailing list would be a great idea to keep.

11. Did you attend the project seminars? Were they helpful? Any suggestions for additional seminars and/or improvements to existing seminars?
The project seminars were great. The only draw back was that these were done usually after the plan was due. This required me to learn more about the project before just coding. Unfortunately my team always spent a lot of time on research before we started coding so the seminars were not as helpful for my team.

12. Have the mailing list and the course wiki been useful in understanding and completing the projects?
The course wiki helped a lot when we needed the files in the documentation section. The mailing list however was a little hard to respond when most of the questions are very complicated. I think if the last project was actually on kernel newbies I think that the mailing list on their would be great or help the course wiki.

13. How much time have you been spending on the projects? Have you been able to finish the projects?
We spent between 10 – 15 hours on each project. We choose not to divide the project because of the complexity, instead we choose to work on it as a whole. This did take a lot longer then it would have if we did divide it but group understanding helped reduce coding errors and also gave us all a better understanding.

14. Have the projects been graded fairly?
Yes. We didn't have any problems with the grade.

15. Other comments/suggestions:
I wish we could have written a driver!

Response #32
1. Have the project definitions/expectations been clearly stated?
While I do the projects in this term, the project definitions were clearly stated. Also, there were project seminar for most of the projects. It was very helpful even if some definitions are not clear.

2. Have you had access to appropriate tools for completing the projects?
Each team were assigned distinct the virtual machine for implementing, testing, and debugging. The virtual machine was always working and there was no problem with accessing the virtual machine.

3. Has the level of difficulty been appropriate for this course?
Since this course is the advanced course of operating system 1 and 400 level course, I expected that this course would be very difficult. As I expectation, the course was appropriately difficult, but it was not very difficult to approach the problems.

4. Were you sufficiently prepared for the level of C programming required for these projects?
I've never used C program before I came to OSU and I've used C program in Data Structure class last year. However, learning data structure was enough to solve this course's problems because I've learned many types of data structures such as linked list, queue, and stack. Therefore, I think I was well prepared for this course.

5. Have the projects helped you to understand operating systems concepts?
I would say this course was definitely helped me to understand operating system concepts because I have accessed to linux kernel and compiled the new kernels. It was totally different from just using operating system. Therefore, modifying the actual operating system was very helpful.

6. Is Linux an appropriate basis for these types of projects, or would you prefer using a proprietary model (e.g., Windows) or a small “academic” model (e.g., Minix)?
Linux is very appropriate for these types of projects because Windows or some other operating system is hiding their source code, but Linux does not. Therefore, Linux is very practical for actual kernel development.

7. Have the projects helped you to understand the Linux kernel?
Definitely helped. Since the first assignment, our group has been making different types of new kernels and we have applied the kernel into the virtual machine. It was more helpful than just understanding the concepts of Linux kernel.

8. Have the projects increased your interest in career work in Linux kernel development or other open-source projects?
As I told my one of group members, developing Linux kernel was very exciting. Honestly, one of my dream is building my own operating system. That is why I have tried to understand all the projects very clearly and the project topics increased my interest in Linux kernel development.

9. Do you prefer projects in several different areas of the kernel or would you prefer projects that go into depth on a single piece of the kernel (e.g., the I/O system)?
I would prefer projects in several different areas of the kernel because if we have experiences with different areas of the kernel, we can think about what kind of kernel developing I would want to do.

10. Have you asked the TAs for help on any of the projects? Have they been helpful?
We asked the TAs a lot for helping out projects. Especially, TA Tony was very nicely helpping our project every times. He gave us some ideas to approach the problems.

11. Did you attend the project seminars? Were they helpful? Any suggestions for additional seminars and/or improvements to existing seminars?
Even if I don't understand the project definition, project seminars helped me a lot because the seminars actually covered and explained about actual code so that we can implement more easily.

12. Have the mailing list and the course wiki been useful in understanding and completing the projects?
The mailing list was very useful when we have problems with implementation or understanding the concepts because classmates and TAs replied quickly. However, we didn't use the student wiki that much since there were not enough posts that I can refer.

13. How much time have you been spending on the projects? Have you been able to finish the projects?
Actually, I have never spent this much time for finishing projects in OSU. It is because most of the projects required compiling the kernel and it always took some time. Also, debugging the kernel code was more difficult than normal programming such as java and C programming. However, our group could finish each project on time since we always started working earlier.

14. Have the projects been graded fairly?
Grading was fair because even though we worked together as a group for the projects, we can have different grades by writing the review document and contribution agreement. However, I wish that if there are more detailed policy for the contribution, students who actually didn't do anything for the projects would get their grade as much as they contributed.

15. Other comments/suggestions:
Personally, CS 411 was really good and helpful for me because this course was very clearly scheduled and the objectives were very interesting. However, it would be better if there are some specific policies for teamwork since 4 students should work together for all the projects as group.
Response #33

1. Have the project definitions/expectations been clearly stated?
   Yes and no. I think that the information provided in the project definitions has been great, however, there has been a lot of stuff, such as vm related issues, which haven’t been provided in the project definition. Overall, I think they’re great.

2. Have you had access to appropriate tools for completing the projects?
   Yes. Through the mailing list, there has been a lot of links and notifications which have been sent around to help us successfully complete the projects.

3. Has the level of difficulty been appropriate for this course?
   I think that the difficulty of this course is ridiculous. However, I feel that this is one of the classes from which I’ve learned the most. I think that if teams were a bit better distributed, this class wouldn’t of been so difficult.

4. Were you sufficiently prepared for the level of C programming required for these projects?
   No. I think that only having touched C in cs261 and cs311 isn’t enough. I think that everyone should be required to take an advance C-programming course (instead of Java).

5. Have the projects helped you to understand operating systems concepts?
   Yes. Having to implement an IO scheduler, create system calls, and hack the kernel, definitely brought me to a closer understanding of how operating systems work. I never really understood kernel space versus user space before this class, however, now it’s crystal clear.

6. Is Linux an appropriate basis for these types of projects, or would you prefer using a proprietary model (e.g., Windows) or a small “academic” model (e.g., Minix)?
   Linux was perfect. It’s very well documented, and there’s a lot of resources from which one can read to gather necessary knowledge.

7. Have the projects helped you to understand the Linux kernel?
   Yes. Before this course, my experience with Linux was close to nothing. After this class, I feel much more comfortable working with Linux, especially with the Linux kernel.

8. Have the projects increased your interest in career work in Linux kernel development or other open-source projects?
   Absolutely. After doing Linux kernel development, I’m now considering changing my career path into Linux kernel programming. The drawback is that I don’t have much “real” experience (outside of school) to help me land a job as a kernel programmer.

9. Do you prefer projects in several different areas of the kernel or would you prefer projects that go into depth on a single piece of the kernel (e.g., the I/O system)?
   Having to touch different pieces of the kernel is what made this class so interesting. Although every project was challenging, I was able to take away a little piece of knowledge, respecting process management, disk io, memory management, etc.

10. Have you asked the TAs for help on any of the projects? Have they been helpful?
    Tony Nyugen was extremely helpful during the projects. It was because of Tony that we were able to clearly understand the requirements and logical concept of what our programs should do.

11. Did you attend the project seminars? Were they helpful? Any suggestions for additional seminars and/or improvements to existing seminars?
    As I’ve stated in my reviews, Rob Hess’s seminars are absolutely necessary if one is to complete the projects. Before most seminars, I would read the book and outside sources, and try to understand the assignment by myself; however, after going to a seminar, would come out with twice as much knowledge and understanding of the project.

12. Have you attended the mailing list and the course wiki been useful in understanding and completing the projects?
    The mailing list were my third knowledge base. Through the mailing list I was able to get answers to a lot of my questions, without having to be stuck for long periods of time.

13. How much time have you been spending on the projects? Have you been able to finish the projects?
    I was able to finish all the projects, with the exception of project 3. On average, I spent about 25-35hrs per project.

14. Have the projects been graded fairly?
    I believe so. My team and I scored 100% on all assignments but assignment 3.
15. Other comments/suggestions:
   As I discussed with Dr. Paulson, my team experience was very bad. Two of my partners did little to no work, which made all the projects extremely stressful and long. If there was some way of switching teams, that would be a huge help for those who feel stuck in a team of lazy people.

Response #34
1. Have the project definitions/expectations been clearly stated?
   I feel as though the project definitions were very well stated. What was unclear I saw as an opportunity to complete as we saw best fit. It made it feel like a real world project to some degree rather than just an assignment.

2. Have you had access to appropriate tools for completing the projects?
   The only tool we really needed was a way to access the virtual machine, which worked well. Once the virtual machine was started we were capable of installing whatever software we wanted or needed.

3. Has the level of difficulty been appropriate for this course?
   I had actually anticipated the project being more difficult, and feel it could afford to be more difficult. However I understand the level of difficulty considering its a core CS class. Since I took it as an elective, I would have preferred it more difficult though.

4. Were you sufficiently prepared for the level of C programming required for these projects?
   I was very well prepared for the level of C programming involved, I actually found that the actual code writing was the simplest aspect, and the understanding of the algorithms we needed to implement was the biggest barrier.

5. Have the projects helped you to understand operating systems concepts?
   The projects didn't help with the overall understanding of operating systems as they were application specific. They did help with the understanding of the project subject material however.

6. Is Linux an appropriate basis for these types of projects, or would you prefer using a proprietary model (e.g., Windows) or a small “academic” model (e.g., Minix)?
   My brother graduated from OIT and they built their own operating system from scratch starting with a building the kernel. I happen to favor that approach as it requires a thorough understanding of what the kernel does, but I feel it may be too much to fit into one course (I'm not sure how they did it in that class)

7. Have the projects helped you to understand the Linux kernel?
   I would say the projects helped a little bit, but the projects were so specific that they didn't deliver an overall understanding of operating systems in general

8. Have the projects increased your interest in career work in Linux kernel development or other open-source projects?
   I've always been interested in open-source projects, and this class more or less fueled that. I think it could have been better had we actually submitted some sort of patch or fix for something, and given us the feeling that we've actually done something, but I realize thats quite difficult

9. Do you prefer projects in several different areas of the kernel or would you prefer projects that go into depth on a single piece of the kernel (e.g., the I/O system)?
   I would have preferred the projects to be more general regarding the kernel as a whole, delving into a single piece of the kernel doesn't really help with general understanding

10. Have you asked the TAs for help on any of the projects? Have they been helpful?
    We haven't needed significant help on any of the projects, we seemed to understand them and implement them fairly easily.

11. Did you attend the project seminars? Were they helpful? Any suggestions for additional seminars and/or improvements to existing seminars?
    I personally only attended one of the project seminars, but I felt as thought the information retrieved from those seminars helped with completing the projects significantly.

12. Have the mailing list and the course wiki been useful in understanding and completing the
13. How much time have you been spending on the projects? Have you been able to finish the projects?
I feel I spend an average of about 8 hours on each of the projects, and we were able to finish all of them without problem.

14. Have the projects been graded fairly?
We never ran into any issues with the grading, we seemed to get the majority of our points, and if we lost points, it was easy to see why we lost these points.

15. Other comments/suggestions:
If possible, having the students write an operating system from scratch I think would make for a very good class. The operating system doesn't have to be as good as Linux, but something to really give students an idea of what it takes to make an operating system

Response #35

1. Have the project definitions/expectations been clearly stated?
   Yes, but it would have been more convenient if each project definition had web links to useful locations in the Linux source – so that it would be organized “go-to” place for getting info on the project.

2. Have you had access to appropriate tools for completing the projects?
   Yes, the email discussions were an especially useful resource for the projects.

3. Has the level of difficulty been appropriate for this course?
   Yeah, it would have been nice if the projects could have been broken down into smaller pieces that were due earlier to make sure we were keeping up with things—I know that plan-of-attacks were probably there to serve that purpose—but if there was a way to break it down further and have parts of the code due ahead of time, I think that would have been helpful (not sure if that’s very possible though).

4. Were you sufficiently prepared for the level of C programming required for these projects?
   Yes, although I’ve done a lot of C programming before OSU and also in ECE473, so I may not be very representative of the class.

5. Have the projects helped you to understand operating systems concepts?
   Yes. I did hope to write programs that have more “tangible” and interesting results and output though. That may just be how it goes with kernel programming. Maybe it’s just me, as an ECE, but I really get excited about something when I can really see what it does when I’m done with it. Again, that may not be possible in this context.

6. Is Linux an appropriate basis for these types of projects, or would you prefer using a proprietary model (e.g., Windows) or a small “academic” model (e.g., Minix)?
   Linux is definitely best. I would definitely not change this.

7. Have the projects helped you to understand the Linux kernel?
   Yeah, I feel like the projects all demonstrated different important parts of the kernel.

8. Have the projects increased your interest in career work in Linux kernel development or other open-source projects?
   To a degree… The kernel used to be a magic black box to me, but now I understand it a lot more – I feel like I would at least know where to start if I ever wanted to spend time on an open source project.

9. Do you prefer projects in several different areas of the kernel or would you prefer projects that go into depth on a single piece of the kernel (e.g., the I/O system)?
It seems best to have a project that focuses on one specific area – it’s confusing enough to try to figure out how just one area works.

10. Have you asked the TAs for help on any of the projects? Have they been helpful?
   I haven’t asked them anything, but they’re involvement on the email discussions was very helpful.

11. Did you attend the project seminars? Were they helpful? Any suggestions for additional seminars and/or improvements to existing seminars?
   I did attend them and they were helpful. Some of the seminars seemed somewhat disorganized. It would have been better if the TAs were a bit more organized and prepared – but they definitely were helpful.

12. Have the mailing list and the course wiki been useful in understanding and completing the projects?
   I barely even looked at the course wiki, but the mailing list was invaluable.

13. How much time have you been spending on the projects? Have you been able to finish the projects?
   I usually put in about a day’s worth of work into the project. The projects often seemed simple, but then I ran into lots of errors and confusion. I wish I had gotten started early (and maybe have been forced to do so in some way…).

14. Have the projects been graded fairly?
   Yeah, I feel like the grading was done very well. These projects are confusing and the errors are sometimes very inexplicable, so I really appreciated the graders looking over the code and giving partial credit.

15. Other comments/suggestions:
   I feel like I’ve said it all. I really think it’s great that we got to work on the actual Linux kernel in this class.

**Response #36**

1. Have the project definitions/expectations been clearly stated?
   In general I have always found the definitions and expectations to be clear. It would be nice if we could get the scoring definition earlier as that often adds a good bit of clarity to what is expected of us.

2. Have you had access to appropriate tools for completing the projects?
   In as much as the tool actually exist, yes we have had access. Things like a better kernel debugger would make life worlds easier but may not exist anywhere.

3. Has the level of difficulty been appropriate for this course?
   Yes, though I feel like the projects could have been conceptually more difficult. I feel like I learned a lot more about Linux than I did about operating systems.

4. Were you sufficiently prepared for the level of C programming required for these projects?
   Yes but I went to PCC for my 161/162 classes and it is all taught in C++.

5. Have the projects helped you to understand operating systems concepts?
   As I mentioned earlier, they helped me understand Linux, not so much operating systems. They felt too conceptually easy to really be useful for learning about operating systems but were very difficult in terms of trying to understand the Linux setting.

6. Is Linux an appropriate basis for these types of projects, or would you prefer using a proprietary model (e.g., Windows) or a small “academic” model (e.g., Minix)?
   I think I would prefer to use a Windows model because I would expect the coding to be of higher quality and the reference to be better using a proprietary system like Windows. If that is not then case then I would rather stick with Linux.

7. Have the projects helped you to understand the Linux kernel?
   Yes, that I feel is the main thing that the projects have accomplished. I feel fairly confident that I can find my way around the kernel now.
8. Have the projects increased your interest in career work in Linux kernel development or other open-source projects?
No, they have confirmed that this is in fact something I want to stay far away from. I want to work with user interfaces and this is very very removed from that.

9. Do you prefer projects in several different areas of the kernel or would you prefer projects that go into depth on a single piece of the kernel (e.g., the I/O system)?
Drivers, I would love to be able to write drivers. If the whole quarter focused on drivers, I would be happy. I feel like I may want to interface things with an OS and still have no real idea of how to do that.

10. Have you asked the TAs for help on any of the projects? Have they been helpful?
Didn't ask, can't rate.

11. Did you attend the project seminars? Were they helpful? Any suggestions for additional seminars and/or improvements to existing seminars?
Yes, super helpful. I would love to have these BEFORE the plans of attack are due. The plan almost always changed dramatically after these presentations.

12. Have the mailing list and the course wiki been useful in understanding and completing the projects?
Mailing list has been very helpful. The wiki seems to be too dated to really be useful except for the first project where we found it very useful.

13. How much time have you been spending on the projects? Have you been able to finish the projects?
We have finished all of them. We probably spent an average of 10-15 hours per project, except for project 3 where we spent probably close to 30 hours. We may have been fighting bugs that were not our fault, I don't really know.

14. Have the projects been graded fairly?
Yes, no complaints here.

15. Other comments/suggestions:
Nothing comes to mind. I would like to maybe see some more information on Windows and Mac throughout lectures. I feel like I really only learned about Linux when I will likely be developing software for Windows in the future.

Response #37

1. Have the project definitions/expectations been clearly stated?
For the most part, yes. Any unclear parts were usually worked out on the mailing list.

2. Have you had access to appropriate tools for completing the projects?
Yes, all that's really needed is a version control system, text editor, SSH and compiler.

3. Has the level of difficulty been appropriate for this course?
Yes, the course is fairly difficult but that is expected from a 400-level class. Mostly it just requires a time investment.

4. Were you sufficiently prepared for the level of C programming required for these projects?
Yes, I had sufficient experience with C to be prepared for the class.

5. Have the projects helped you to understand operating systems concepts?
Yes, each project covered a single concept in detail (I/O scheduler, memory management, etc)

6. Is Linux an appropriate basis for these types of projects, or would you prefer using a proprietary model (e.g., Windows) or a small "academic" model (e.g., Minix)?
Linux is a good choice for the projects. It is large and complex, but has a strong community and is well documented. Furthermore, using Linux allows students to gain some "real-world" experience from the class. Something like Minix might be a good choice for learning certain concepts that would require overly complex hacking of the Linux kernel, or for a project where students must implement a subsystem from scratch. I would not try to base the class on Windows.

7. Have the projects helped you to understand the Linux kernel?
Yes, the projects interface directly with some of the most important aspects of the kernel, so it's hard not to gain an understanding.

8. Have the projects increased your interest in career work in Linux kernel development or other
open-source projects?
They haven't really increased or decreased my interest. I'm interested in low-level C programming, but I would be fine working on either a proprietary or open-source project.
9. Do you prefer projects in several different areas of the kernel or would you prefer projects that go into depth on a single piece of the kernel (e.g., the I/O system)?
I like the in-depth projects because I think they really help to give a full understanding of a particular aspect of the operating system.
10. Have you asked the TAs for help on any of the projects? Have they been helpful?
I have not asked the TAs for help.
11. Did you attend the project seminars? Were they helpful? Any suggestions for additional seminars and/or improvements to existing seminars?
I attended all the project seminars and they were EXTREMELY helpful. No suggestions except that you should keep having them, they made doing the projects much easier and make much more sense.
12. Have the mailing list and the course wiki been useful in understanding and completing the projects?
The mailing list has been very very helpful and prevented me from spending a lot of time on things that didn't work at first. The course wiki was also helpful - particularly the sections on compiling the kernel, booting the kernel and loading modules which I referred back to frequently.
13. How much time have you been spending on the projects? Have you been able to finish the projects?
I have spent an average of 6 hours on each project. Unfortunately I have a tendency to get started late and this has affected our ability to complete the projects. Usually they are "mostly" working, but with one or two bugs.
14. Have the projects been graded fairly?
Yes, I have no complaints about the grading.
15. Other comments/suggestions:
Because of the time consuming nature of the class, it's probably best taken with a schedule that is fairly open otherwise. If I had been less busy this term I probably would have gotten more out of the class. (This is more a suggestion for students taking CS 411 than for the development team.)

Response #38
1.1 Have the project definitions/expectations been clearly stated?
Yes, but they've been ambiguous enough to make us research. The definitions sent us in the right direction, but the book and Google proved to be much needed resources.
1.2 Have you had access to appropriate tools for completing the projects?
Yes. The guest seminars were very helpful. Plus, everything's on the internet these days.
1.3 Has the level of difficulty been appropriate for this course?
I think so. But I've talked with a lot of other students who have expressed frustration at the difficulty of the projects. Our team seemed to be pretty clever, though. So it was okay.
1.4 Were you sufficiently prepared for the level of C programming required for these projects?
Sort of. Personally, I think OSU needs to introduce engineering students to pointers before they get to CS 261. I don't know how the majority of engineers feel on the topic, but I think 151 may have been beneficial.
1.5 Have the projects helped you to understand operating systems concepts?
Yes. Things make the most sense to me when I not only have to study, but also implement computer science concepts.
1.6 Is Linux an appropriate basis for these types of projects, or would you prefer using a proprietary model (e.g., Windows) or a small academic model (e.g., Minix)?
Linux is awesome. And it may introduce some students to open source, which I think is always a good thing.

1.7 Have the projects helped you to understand the Linux kernel?
Yup. I'd always wondered what it was. Some friends I looked up to in high school would talk about recompiling their kernel. I mostly thought about popping kernels into popcorn. But now I understand. It's a good feeling.

1.8 Have the projects increased your interest in career work in Linux kernel development or other open-source projects?
It's made the open source community more accessible. This is the first class I've been in where the mailing list was actually used. It was nice to get experience in a mail intensive environment, funny as that may sound.

1.9 Do you prefer projects in several different areas of the kernel or would you prefer projects that go into depth on a single piece of the kernel (e.g., the I/O system)?
Several different areas is the way to go. Let's assume everyone has a favorite aspect of the kernel. Let's also assume that there are 5 projects over the course of a term. I'd say it's better to please everyone a guaranteed fifth of the time than to over-please a fifth of the people. Just sayin'.

1.10 Have you asked the TAs for help on any of the projects? Have they been helpful?
Actually, no. Our group was pretty good at seeking out answers among ourselves. I would guess they're helpful people, though.

1.11 Did you attend the project seminars? Were they helpful? Any suggestions for additional seminars and/or improvements to existing seminars?
I liked the seminars. I really appreciated having a whole session dedicated to the project I was about to work on. They helped me understand the projects conceptually.

1.12 Have the mailing list and the course wiki been useful in understanding and completing the projects?
Yes. The wiki was especially helpful when doing repetitive but specific tasks, such as SVN branching and VMware use.

1.13 How much time have you been spending on the projects? Have you been able to finish the projects?
We've finished them. I'd say the time we've spent on the projects has varied from about 10-20 hours a piece.

1.14 Have the projects been graded fairly?
Yes. At least, our group never got docked for something that seemed out of line.

1.15 Other comments/suggestions:
I find I learn the least from PowerPoint-driven lectures. In CS 311, we often had (somewhat) interactive terminal sessions projected in front of the class, which gave more of a hands-on feel.

Response #39

1. Have the project definitions/expectations been clearly stated?
The definitions always seemed stated clearly enough on the initial read through, but once we would start working on the project bits and pieces seemed to be missing. Most of the time I always assumed that the missing bits were left out intentionally.

2. Have you had access to appropriate tools for completing the projects?
The only new tool that I was introduced to in this course that wasn't used in CS 311 was the VM, which I had good access to.

3. Has the level of difficulty been appropriate for this course?
The level of difficulty was very high for me and most of my group, but for some groups these projects didn't too bad for. Sometimes I felt like the projects were made to be a little more difficult than they needed to be, but maybe the moral of the story is that Linux kernel programming is difficult.

4. **Were you sufficiently prepared for the level of C programming required for these projects?**
   I made it through CS 311 and did OK, so I feel like the level of C required for these projects was a little more complex than in 311, but it was appropriate.

5. **Have the projects helped you to understand operating systems concepts?**
   They really have. These projects really helped me understand what is happening under the hood of an operating system. They made me think of, and understand the concepts that I had never really thought in depth about before taking this class.

6. **Is Linux an appropriate basis for these types of projects, or would you prefer using a proprietary model (e.g., Windows) or a small “academic” model (e.g., Minix)?**
   I just took CS 419 which is all about open source software, and Dr. Budd made it clear that there is a movement being made to work open source software into the curriculum. Using Linux for CS 411 is a perfect way to introduce students into OSS development. However, if it was concluded that using something like Minix would help teach this class more effectively, than maybe that would be the way to go.

7. **Have the projects helped you to understand the Linux kernel?**
   Until taking this class, I had zero previous experience with the Linux kernel, so these projects compose around 80% of what I know about the Linux kernel.

8. **Have the projects increased your interest in career work in Linux kernel development or other open-source projects?**
   I can't say that these projects have increased my interest in career work in Linux kernel development, but I can say that if someday I ended up having a career like that, I would be thankful that I took this course.

9. **Do you prefer projects in several different areas of the kernel or would you prefer projects that go into depth on a single piece of the kernel (e.g., the I/O system)?**
   I think it would be more effective to focus on several different areas of the kernel to help students understand more of the kernel instead of a single piece really well.

10. **Have you asked the TAs for help on any of the projects? Have they been helpful?**
    My group went to the T.A.s for help on ¾ of the projects. He was very helpful, but the problem was that these projects can be implemented in so many ways that when he'd look at our code, and still not understand what our problem was because our code looked good, we'd still be stuck.

11. **Did you attend the project seminars? Were they helpful? Any suggestions for additional seminars and/or improvements to existing seminars?**
    I attended all of the seminars, and they were extremely helpful. Without the seminars, I feel like hours and hours of more work on our end would have been added to the projects. I think it would be great if there was a seminar for each project. The format used on the current ones is perfect.

12. **Have the mailing list and the course wiki been useful in understanding and completing the projects?**
    The mailing list and course wiki have been invaluable. Without these resources to fill in the gaps that the project descriptions leave, and the questions that arise when working on the projects, I feel like my team never would have gotten through any of them.

13. **How much time have you been spending on the projects? Have you been able to finish the projects?**
    The thing that bothers me most about the projects is how long they take to complete. I've spent at least 20 hours on each project, but it was normally much more than that. Debugging the kernel is one of the most irritating and time consuming things I've ever had to do. Unfortunately my team wasn't always able to finish the projects on time, but you can't say it was because we didn't try hard enough, or put enough time into them.

14. **Have the projects been graded fairly?**
    The projects definitely have been graded fairly, and I appreciate the fact that I always felt like the graders were on our side, and weren't out to get us. These graders seemed to understand how difficult these projects are, and how much time we put into them.
15. **Other comments/suggestions:** It shows how much work you guys have put into making these projects for us, and you've done a good job.