

Programming Assignment #5 - Team Project

A useful device driver or other kernel module

Value:

(See the **Grading** section of the *Syllabus*.)

Due Date and Time:

(See the **Course Calendar**.)

Summary:

This is your final project. You will be writing a new or improved device driver, or possibly another kernel module, working on a team of one to three people. Ideally, you should propose a project that interests you.

Objectives:

- Work on a real device driver or other kernel module, providing useful functionality that did not exist before.
- You will need to agree on a set of more detailed objectives within the team and obtain approval of those objectives by the instructor. In general, the criteria for a project are:
 - It performs some new useful practical or research purpose.
 - It requires you to write new code that executes in the kernel.
 - The new code requires you to demonstrate ability to use significant elements of the kernel internal programming interfaces and concepts covered in the course.
 - It appears likely that a team can complete it, or at make least enough progress to be evaluated, in four (4) weeks.

Choice of Project & Criteria

You are responsible for choosing project and a team. See Final Project Ideas document for a list of possible projects.

Tasks and Due Dates:

Task	Due Date
1. Form a team, and tell me who your teammates are. Be certain that the team members have compatible schedules, since you will need to communicate a lot. Meet and decide how you will first divide the work. Plan on revising this work breakdown repeatedly as you make progress and understand the task better.	By: Friday March 8 th What: Teammate names How: Email

<ol style="list-style-type: none"> 2. Read the relevant references 3. If you are doing a driver, you should get the device up and running with whatever other driver may exist for it (either an existing Linux driver, or if there is no Linux driver then a Windows driver). 4. Read through the existing driver code, if such is available, and design your changes; otherwise, if you are starting from scratch, design your driver. 5. Write up an initial 1-page proposal covering the following points <ul style="list-style-type: none"> • Names of group members • Project description • Background and related work • General project design • Expected challenges 	<p>By: Tuesday, March 19th</p> <p>What: Design document</p> <p>How: Email</p>
<ol style="list-style-type: none"> 6. Demonstrate that you have the older driver and demo program (if any) working in class. In addition, give a 10-15 minute presentation of your team's plans and any progress in class that day. It would be best if you have Web-readable notes to go with your presentations. 	<p>By: Tuesday, March 26th</p> <p>What: Initial project presentation</p>
<ol style="list-style-type: none"> 7. Develop a working prototype, with at least some of the (new) functionality you have planned. Give a team 10-15 minute presentation on this in class. 	<p>By: Thursday, April 9th</p> <p>What: Midway project presentation</p>
<ol style="list-style-type: none"> 8. Spend the part of the course refining your driver and writing up the documentation. Plan on giving an extended team presentation and demonstration in class on the week before finals. Make sure your code and documentation are available to the instructors. 	<p>By: Last week of class</p> <p>What: Final extended project presentation, code, journal, documentation</p> <p>See "Final Delivery Method" section below</p>

Advice:

- I recommend that you follow an incremental and iterative development approach on this assignment, as on the others. That is, you should start with an overall design of where you want to go with the driver, but develop it in stages. You will produce successive "releases" of the driver. Each release will provide a bit more functionality. This way you can code, test, and debug in small chunks. That way you should always have something that is at least partially functional, and will be continuously improving it. By coding and testing in stages you verify that you understand what you are doing before you have written a lot of new code. Inevitably, you will discover that you made some mistakes in your early design, but you will have a chance to revise your plans before you have written out a huge amount of code, and so reduce the amount of rewriting you need to do. If you run out of time, you should always have something that is at least partially functional to turn in.

- For this incremental approach to work, **source code control is critical**. If an incremental changes turns out to be a bad idea, you need to be able to roll back to the previous version. Plus, using git with branches will help me evaluate the amount of work completed, even if all branches are not fruitful.
- Take advantage of having a team by doing code design and walk-throughs. One person writes down a piece of the design, and later of the code. The others read it, ask critical questions, and make suggestions for improvements. You then reverse roles.
- Dividing up the work among the team members will be a challenge. To make effective progress you need to either work very closely together, or split up the work into pieces that are sufficiently independent that they can be done in parallel. One tempting way is to split off the documentation from the coding, and the development of tests from the development of the device driver itself. I recommend against this, at least at first. If one person does all the coding and debugging, that person will do most of the learning. That person will also make mistakes that go undetected because only one pair of eyes and one mind are looking at the code critically. A person who writes documentation only will risk not writing it correctly, from lack of understanding. I would recommend doing nearly everything "hand in hand" up to a point where you have a working framework and everyone understands it. You can then assign different individuals to add more methods, and more options/parameters to the methods. For the documentation, I suggest you together develop an outline, have each person draft a portion of the document, and then each swap pieces to do the editing, until everyone has approved every line of text.
- As in the previous assignment, you are free to work out the unspecified details, according to your own best judgment, but remember that the **quality** of such decisions will be taken into account in the assessment of your work. Always keep in mind the objectives of the assignment. You are trying to make a *useful* device driver, good enough that other people will use it. This means not just good in functionality, but easy to compile and install, and compatible with all kernel versions currently in use. When you make design choices, choose in the direction of providing greater efficiency, functionality, portability, and convenience for users. There may be ways you can reduce the amount of work you have to do, at the expense of some other consideration. If you chose that path, do so with the understanding that you may also reduce your score. If you have doubts or questions, ask me. If you bring up issues that I think are of general interest to the class I will post them on the course web pages so that other students may also benefit.
- Get started right away. You don't have long to do it.

References:

- Web pages from previous student projects
- Project ideas presentation

Final Delivery Method:

1. Please follow instructions given in class for details, including turn-ins and reviews of intermediate results at the end of each week.
2. Your team will be expected to give a final presentation and demonstration of your project in the last two days of classes.
3. Each person is required to individually provide me with a signed sheet of paper indicating your estimation of the percentage contribution of each team member to the project, something like this

	Your Name	Team Member Name	Team Member Name
Percentage Contribution			
Your signature:			

4. Before the final grade is turned in, each team is required to turn in a *tar* archive of the team web page, including:
 - a. The names of the team members
 - b. A description of your project goals and accomplishments.
 - c. A journal of your progress
 - d. The code that you wrote, including any test programs.

Assessment:

- Your work will be judged according to quality of function (including the number of the deficiencies of the older driver, if any, that you use as a starting point), quality of design, quality of coding, quality of testing, and quality of documentation.
- Quality of coding will include (but need not be limited to) consideration of simplicity, readability, portability, efficiency, and robustness.
- The documentation includes not just comments in the code, but also the user-level documentation (how an application uses the driver or other kernel modules that you produce).
- If you work on a device driver, your design should provide access to *all* the functionality of the hardware, and also make the device as convenient to use as you are able.
- You may be asked to demonstrate the installation and compilation of your software "out of the box" on an plain "vanilla" Linux distribution.