

Programming Assignment #5

A useful device driver or other kernel code

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 other kernel code. Ideally, you should propose a project that interests you.

Objectives:

- Demonstrate your ability to design and write kernel code.
- Work on a real device driver or other kernel code, providing useful functionality that did not exist before.

Criteria

- 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.
 - Small enough to complete or at make least enough progress to be evaluated, in approximately four (4) weeks.

Tasks:

- Choose a project and obtain approval of instructor for said project.
- Initial presentation describing your project.
- Mid-term presentation discussing progress on project.
- Final presentation describing your accomplishments.
- Write a report/documentation discussing what you have accomplished.

Final Delivery Method:

- Please follow instructions given in class for more details.
- You will be expected to give a final presentation and demonstration of your project during the week of classes.
- Before the final grade is turned in, you are required to turn in the following:
 - A **report** describing your goals and accomplishments, which includes:
 - *Documentation* on design and use of your kernel code.

- Achievements
 - What goals were reached?
 - If some goals were not reached, what were the obstacles/difficulties?
- Future Work
 - What could be done if you continue the project?
- The **code** that you wrote/changed, including any test programs.

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.
- 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 compatibility with recent kernel versions. 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 the instructor or TA.
- Get started right away. You don't have long to do it.

References:

- Web pages from previous student projects

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 functionality that you produce).

- You may be asked to demonstrate the installation and compilation of your software "out of the box" on a plain "vanilla" Linux distribution with a kernel version of at least 3.10.