

Final Project

Due	No Due Date	Points	0	Submitting	a file upload
-----	-------------	--------	---	------------	---------------

Introduction and Deliverables

For the final project for CIS 1910, you will need to get your hands dirty with some kind of Linux/Unix project. This means you have a lot of flexibility! We provide some suggestions below but feel free to be creative. There is no strict guideline for how big the project must be, but a good rule of thumb is that it should have a larger scope than any of the previous homework assignments. This is a chance for you to show off some of the skills you've learned in this class. This may include doing some additional research to accomplish your goals. You are allowed to work with a partner for this project. The deliverables for this project are:

- Proposal (due April 11th)
- Repository (due April 25th)
- Write up (due April 25th)
- Presentation (presented in class April 25th)

Project Ideas

You are allowed to come up with your own project proposal, but here are some suggestions to get you started if you're stuck for ideas. Note that you are not required to use a language you learned in this class like Bash, but you should apply the skills from CIS 1910 in some way.

Option 1: Build a tool you will find useful

We've seen tons of different command line tools available to us. Now it is your chance to write your own! Here are some potential tools you could create:

- An emacs mode or vim plugin
- Dropbox-like backup tool
- Calendar/to-do list notification tool
- Build automation tool
- Tool for managing your data: photos, music, videos, etc.
- Framework for managing your coursework or job search

You can even expand upon some of the topics brought up in previous assignments! Want to turn that simple library system into a legit more robust database? Maybe take a look at learning MySQL (already installed on Eniac) for revamping the system. How about that autograder assignment? Think you could build a system for submitting and grading assignments?

While this course is not a web programming course, part of the goal is providing practice learning new skills on your own. You could consider learning how to run and host a small web application using your personal webpage on Eniac. Some instructions from CETS on how to setup your personal webpage can be found [here](#), specifically under the Public HTML section.

There are *tons* of options here! If you are feeling stressed about coming up with a project, please reach out to course staff for help. This is intended to be a fun and useful final project.

Option 2: Contribute to an open source project

Linux has a very close relationship with Free and Open Source Software (FOSS). One option for the final project is to contribute to an open source project. You can browse for projects [here](#) ➞ or [here](#) ➞. Some ways you can contribute might include:

- Add a feature
- Fix a bug that has been reported on the bug-tracking forums
- Package the project for distribution on various OSs (e.g. .deb for Debian-based systems, port to OSX, etc)
- Add documentation, examples, a tutorial, etc.
- Add support for Wine to run windows executables on Linux machines

Again, you are free to work with whatever tools or programming languages you like. Make sure to read up on guidelines for contributing to a project, as well as the project specific documentation.

Option 3: Investigate security issues

Break something or fix something that's broken! There are tons of historic security vulnerabilities you can investigate as well as ones still around today. From physical security to cryptography, you can find interesting projects to build or exploits to try. One thing you could do is research buffer overflow attacks and create a project demonstrating how this security vulnerability works. The article [Smashing the Stack for Fun and Profit](#) ↓ may be of interest to you in learning about this attack. You could also research cryptographic techniques and build a tool around demonstrating them.

Option 4: Build your own kernel

This is not an operating systems course, but if you are feeling particularly motivated or interested you could take a crack at creating your own basic OS. After all, it was Linus Torvalds' interest in the new intel i386 processor in 1991 that caused him to create a kernel that eventually developed in Linux! You most certainly will not get a fully working OS in four weeks, but even attempting to make one is a valuable learning experience. A real OS course will give you a more structured learning environment, but there are tons of OS Dev guides out there such as [James Molloy's kernel development tutorials](#) ➞ to get you started.

Project Assessment

The goal of this final project is to give you more hands on practice and encourage you to explore the variety of things you can do with Unix/Linux. You will get out of it what you are willing to put into it. That being said, we still have to evaluate your work in some capacity. Each deliverable should meet the following criteria:

Proposal

- Group members (at most two)
- Description of project
 - Specific components and milestones to complete
- Background on the tools or design of your project
 - If working on a program, what similar tools exist and what are their plusses and minuses?
 - If contributing to an open source project, what is the project about and what contribution will you make to the project?
 - If working on security, do some research about security bugs

Repository

- Must use version control
 - Commit often with [quality messages](#) ➞
- Include a README.md
 - Describe what the project does and how to use it
- Write modular code
- Document with comments
- If applicable, use build tools like Make

Write up

- Group members (at most two)
- Description of your project
 - Specific components
 - If applicable, contributions by student
 - How to use your code (see README.md above)
 - Link to GitHub
- Background on the tools or design of your project
 - See proposal requirements
 - It is okay if your tool implements fewer features than a similar existing tool!

Presentation

- Short (5 minute) presentation on the last day of class
- Demo your project
- Discuss some of the information from the write up
- Be prepared to answer questions

Overall Grade

You can receive one of three grades for the final project:

- Satisfactory (S): For projects that meet all the above criteria for each deliverable and demonstrate sufficient scope and understanding of material; full credit
- Not yet satisfactory (N): For projects that do not meet all the above criteria for each deliverable or projects that do not sufficiently explore and expand upon material covered in the course, i.e., projects that are similar in scope to a standard homework assignment; partial credit
- Unassessable (U): Did not turn in anything; no credit

We recognize that this kind of grading scale may be unfamiliar to you and seem arbitrary. In general, if you put in a good faith effort ***you will receive full credit for the final project***. If you are concerned at all about the scope of your project please reach out to course staff sooner rather than later to discuss. Furthermore, you must turn in ***something*** for the final project in order to receive a passing grade for the course.

Submitting

The write up (which should include the link to your public repository) should be submitted to this assignment ***on Canvas not on Gradescope***. You can submit the proposal at the assignment found [here](#). There is nothing to submit for the presentation.