CMPSC 400
Operating Systems
Spring 2020

Final Project:
Final Project in Operating Systems

Submit deliverables through your assignment GitHib repository bearing your name. Place source code in `src/` and written work in `writing/`.

Figure 1: There are many smaller parts that make up the larger OS. Some pieces of the OS are responsible for memory management, device-IO, system calls, process management and so on. Each of these parts must be designed specifically and, often, different OS's will have the same types of software to handle similar jobs.

Objectives

To investigate a concept or mechanism that plays an important role or function in an operating system (OS). A report is to be written where the concept and its mechanism are fully explained and may be used as a learning experience for others. Source code is also to be written to complement the work and to demonstrate some specific functionality, in connection with the concept or mechanism. The report and code, combined, will serve to educate others about your investigation and provide an intelligent source of information from which the excitement stemming from concept or mechanism can be easily recognized. Note: This project should contain the same amount of work as about three labs and will not include a presentation.

GitHub Starter Link for Groups

STOP! STOP!
Not everyone will be clicking this link at this time!
Only the team leader will be clicking the link to create the repository!!
`https://classroom.github.com/g/QiSM6o_V`

Creating Your Repository

If you would like work as a group, then the option is open to you. If you would like to work individually, then you will be in a group of your own on GitHub Classroom for this assignment. For group assignments **only one person will be creating the team while the other team member will join that team.** Please form a team of **no more than two people** and select one person to create the repository.

The selected person of the team should go into the link to the lab in the assignment sheet. Copy this link and paste it into your web browser. Now, you should accept the laboratory assignment and create a new team with a unique and descriptive team name (under "Or Create a new team").

Now the other members of the team can click on the assignment link and select their team from the list under "Join an Existing Team". When other team members join their group in GitHub Classroom, a team is created in our GitHub organization. Every team member will be able to push and pull to their team's repository.

To use this link, please follow the steps below.

- Click on the link and accept the assignment
- Once the importing task has completed, click on the created assignment link which will take you to your newly created GitHub repository for this lab,
- Clone this repository (bearing your name) and work locally
- As you are working on your lab, you are to commit and push regularly. The commands are the following.
  - `git add -A`
  - `git commit -m ‘‘Your notes about commit here’’`
  - `git push`

Introduction

This semester, we have have been studying many different parts of operating systems. Although our focus has been on Linux-based OS's, the concepts that we have discussed have translations and equivalencies across seemingly all operating systems such as, Unix, Windows, OS/2, MsDOS, DRDos, and others. For each concept discussed in class, we have generally had some type of demonstration to help us learn more about its design, usage and code. The demonstration allowed us the ability to experiment with some artifact and to ask our own "what-if" questions, which we could answer by simply playing with code.

In this project, you are to investigate a concept from OS's which has not been fully discussed in our own class. Your project will be to explore this concept or mechanism to be able to explain it and to offer a simple coded demonstration of its functionality. In addition, you are to provide a reasonable argument as to why your chosen concept is related (and relevant) to the study of operating systems. This implies a very open-ended project which could concern seemingly any part of OS's. *Your task during your discussion is to help the reader understand why your chosen topic may be found in an OS's final project.* **Please note, this project is to be equivalent to the work of three class labs.**

For your final project, you may work in groups no more than three people. You and your colleague(s) will be given the opportunity to perform some investigative research of your own into a topic of OS's to discover an important mechanism that has strong benefits into the design and development of modern OS's. The mechanism that you choose to research can be any one of the many which are integral some of the topics (listed in Section, "Covered Topics") from our OS course (to give some ideas.) During this research assignment, you are to concentrate on exploring the function of the mechanism to determine how it works and then to prepare a demonstration or simulation of the mechanism using a programming language of your choice. In addition, you are to be able to explain this concept from your work to others.

Covered Topics

Although the actual list of covered topics is much longer, some of the more basic may be found listed below. These other topics (not included in the list) can be found in your class notes or by consulting the textbooks of our course [2] and [1]. As you can easily determine, each of these listed topics actually requires much more time to for an in-depth discovery and study of their mechanisms and to learn more about their benefits to the OS. Unfortunately, due to time constraints, our time was limited and we could not spend the entire class time on only one topic, however, your learning and appreciation of a particular focus in operating system does not have to end so soon.

- Shells
- System calls
- Kernel and user-space functionality
- Processes
- Threads
- Virtual memory management
- Memory pages
- Semaphores

- Schedulers
- Multiprocessing
- Mini operating systems, run in emulators
- And others...

Requirements

You are required to complete the following areas for your project.

- Select a topic that you can argue is relevant to OS's
- Complete a literature review of this topic to enhance your expertise in your report and demonstration.
- Write coherently in clear in meaningful language to be able to educate your reader.
- Complete source code to demonstrate a specific element of of your topic.

Assignment Specifications and Due Dates

- Project ideas Thursday, 9^{th} April by midnight:
  - Begin your project by considering three ideas. For each of your ideas, submit one or two sentences to explain the idea in some detail. You will get some feedback on your Issue Tracker from the instructor.
- Proposal Thursday, 16^{th} April by midnight (at least 500 words):
  - Describe your project, explain some basic background about it and add some citations to support this work. Then, say what you will do in your study and what your code will do to demonstrate a concept. Start by choosing one of your ideas and offer peer-reviewed articles to be used as citations in your argument of its relevance to OS's. Please DO NOT use blogs or non-peer reviewed articles: any article that you bring to your work MUST be peer-reviewed and of good academic standing. Please use Google Scholar (`https://scholar.google.com/`) , or similar, to help you locate and cite academic articles. You are to include least three main references in your proposal document to motivate your project. In this proposal document, outline what your work will entail and what your code will do to demonstrate the concept.
- **Presentation**: None required for this semester due to the college closing as a result of the COVID-19 health concern.
- Progress Report Thursday, 23^{rd} April by midnight (at least 500 words):

  - Here you are to describe what you have been working on so far, and to discuss some of the challenges that you have encountered and how these challenges were met. Be sure to include discussion of issues concerning data, programming and similar for your concept demonstration.

- Full Project Report Wednesday, 1st May by midnight (at least 2000 words):

  - In your report, your topic and its related mechanism are to be fully revealed in *clear and meaningful* writing. The department of Computer Science at our college prides itself on its excellent students who are able to produce exceptional code, in addition to well-written documentation concerning the code. Please take your technical writing very seriously as points can easily be lost for meaningless or sloppy technical reports.
  - The instructor expects that you present the elements of the topic and mechanism clearly and provide details to explain why they are relevant to our course of operating systems. Your report will cover about four or five pages of text (no including the graphics) which **must** contain the following information. In this work, you are also to include screenshots of your coded example and to provide interpretations of the "take-home" message from the demonstration. Since there will be no presentation, you are to provide clear instructions about how your submitted code can be compiled, used and understood, in-light of your project.

  The elements of your report are the following:

  1. An clear introduction to your topic and how or where it applies in OS's
  2. Some background about the concept. Why is the concept relevant?
  3. A detailed discussion of your own work to explore the project. Here, you explain the work and offer learning to the reader about the concept. Introduce and discuss your demonstration, offer screen shots of sample output and guide the reader to understand its output.

- **Code:** Discuss your code. For instance, address how your code works (i.e., how to run it, what parameters are necessary to run it) and what is to be understood about the mechanism or concept by the output of the code.

  Note: The code is to be an executable simulation of the mechanism that you have studied concerning your topic. This code will be a generic and dynamic simulation of the mechanism (accepting inputs and determining outputs). Your program must compute a result and instruct the user as to what it is doing to help the user understand the concept or mechanism of the project. In other words, your coded simulation must do *something* to illustrate some important part of your topic and its mechanism which have some clear connection to OS's. Your demonstration should be addressed in the report using discussion and screen shots. Please be sure to add relevant documentation to your code to help the instructor use it. Also, please make note of whether a Docker container (and which one) will be necessary to run the code.

Grading Rubric

1. **Ideas**: 5 points
2. **Proposal**: 10 points
3. **Progress report**: 25 points
4. **Final report and project implementation**: 60 points

Required Deliverables

This assignment invites you to submit an electronic version of the following deliverables.

1. File: `writing/ideas.md`: Your three ideas to consider for the project. See details above.
2. File: `writing/proposal.md`: Your proposal document where you include your literature review. See details above.
3. File: `writing/progressReport.md`: Your progress report. See details above.
4. File: `writing/report.md`: Your final report. See details above.
5. File(s): `src/*`: The working code for your demonstration as detailed above. Please be sure to provide documentation to be able to use this code.

Honor Code

In adherence to the honor code, students should complete this assignment on an individual basis. While it is appropriate for students in this class to have high-level conversations about the assignment, it is necessary to distinguish carefully between the student who discusses the principles underlying a problem with others and the student who produces assignments that are identical to, or merely variations on, someone else's work. As such, deliverables that are nearly identical to the work of others will be taken as evidence of violating the Honor Code.

References

[1] Abraham Silberschatz, Peter Baer Galvin, and Greg Gagne. Operating system concepts", john wiley and sons inc., 2012., 2015.

[2] Andrew S Tanenbaum and Herbert Bos. *Modern operating systems*. Prentice Hall Press, 2014.