Self Assessment - Fall

Codi Burley

Throughout my time as a student in computer science and a worker in the software, I've seen many new technologies revolutionize the way certain industries are done. This is especially apparent in the web development community, where a huge range of new tools and frameworks are developed and adopted quickly and on a regular basis. When it comes to systems level programming, however, old technologies that have historically been used in an unsafe manner are still utilized. By aiming to create a secure real time operating system in Rust, I feel that I'm giving myself the opportunity to adopt a new technology in an area that I don't have much experience in: systems level programming. This project will give me the chance to improve my skills in an area I know little about while attempting to show that new technologies can be used in systems level programming in order to create more secure real time systems, in a time when more and more real time systems are becoming connected to the outside. My hope that is by the end of the project, even if our operating system isn't complete, we have still shown that Rust is a technology that can and should be utilized for similar projects in the future.

As a computer science student, there have been many skills I have acquired that are applicable to this senior design project. General concepts in computing, and specifically programming, were taught in the beginning of our curriculum (in Computer Science I and Computer Science II). These skills are apply to almost all fields of computer science, and helped to build up the knowledge needed in order to complete some of the more specific courses. One of the more advanced courses I took that apply to this project was Programming Languages. Because Rust is a new language, with new constructs, learning how to utilize and design programming languages has helped greatly in understanding how to use learn and use Rust. Another course that will help in my contributions to this project is Operating Systems. In this course we learned the responsibilities and behaviors of many different types of operating systems. It's easy to see how a course like this would be applicable to a project that relies on the development of an operating system.

Much of my knowledge accumulated while at UC has come from not only the Computer Science curriculum, but also from co-op experience. During my time at GE Aviation as a Design Engineer, I learned the importance of redundancy and the importance of the verification of your work. It's vital that the components of a jet engine are verifiably safe, and this is something that will be important in making a secure system like we intend to do for this project. At Kinetic Vision, I served as a software developer, and learned many good practices for managing a large software project that is in collaboration with many people. The operating system that we intent to create will likely include many components and a very large code base, which will require that the team can collaborate in a manner that is conducive of productivity. My last co-op was at NASA Ames as a Research Engineer. There I learned how to address a problem that does not have a definite answer, and how to do research in general, which will clearly help out project. Additionally, the software I wrote at NASA was built on top of a real time operating system. This gave me some experience with how a real time operating system works.

It is a reasonable question to ask why I chose to work on a project that requires writing an operating system, when I don't in fact have any experience in writing an operating system. In the past I have been very interested in programming languages, and I have always leapt at the opportunity to apply something new in order to try it out. Rust is a language that is new to a field of programming that hasn't had many revolutions as of late, and that gave me motivation to attempt to use it in our project. As of late, I have also been interested in cyber security, along with the members of my group. This gave us a reason to look for a project that could contribute to that field. By choosing a project that requires that we write an operating system, I've also given myself the chance to learn how to a kernel which is something I've always wanted to do, and I'm excited to see what kind of results we can get.

To get started on the project, we plan to utilize the previously written real time operating systems. Through analyzing what other real time operating systems got right, and what they did wrong, we can use open source code that will be useful, and find spots we can improve on. Because writing a real time operating system is a hefty task, it is not fully expected that our operating system will be "production ready" by the end. We expect to have an operating system that is functional, with maybe only a few necessary drivers. We hope to show that Rust can be utilized in order to create an operating system that is safe through the utilization of safe constructs in Rust. The evaluation of our project should come through the comparison of our operating system to other real time operating systems in Rust. We hope to show that ours projects against some vulnerabilities that others have not.