

Our Senior Design project is to create a web app that compiles C code into Intel x86 machine code, and animates the process of running the code. During the animation the user will see registers and stack values updating as the machine steps through each line of code. This project will challenge me in a wide variety of the skills I have developed during my academic journey. It will involve creating a frontend user interface that is visually appealing and usable, sending user data to a server to perform compilation into assembly code, and returning that assembly code to the front end. The animation step will involve implementing logic to determine how the assembly code will update the displayed values, updating values accordingly, and providing smooth animations to show the emulator's progression through the code. Completing this project effectively and on time will also require application of the project management skills I have learned over my academic journey.

I believe that the curriculum I have gone through thus far will be instrumental in helping me complete this project. In the fall of my sophomore year, I took Intro to Computer Systems (CS 2011), which introduced me to the high level concepts of computer architecture, as well as assembly code, how it interacts with memory and registers, and how constructs such as the stack and TRAP tables are utilized, which will be instrumental to my ability to learn the assembly code that our project will work with, and be able to effectively implement the animation of its execution. In the fall of my junior year, I took Computer Networks and Networked Computing (CS 4065), which introduced me to the concepts of networked interactions between computers. The material from this class will provide a foundation for the client-server communication that will need to occur in our project. That same semester, I took Web Design I (AMEC 2050C), and this semester I am taking User Interface I (CS 5167). Both of these classes will be helpful in learning the concepts of what makes an aesthetic and easily usable interface, which will be

important for implementing the front end of our project. Finally, my experience working on projects in Computer Networks as well as the spring of my pre-junior year in Software Engineering (EECE 3093C) will be important experience to draw on for effective project management. Other Computer Science fundamental skills such as Data Structures (CS 2028C), and Design and Analysis of Algorithms (CS 4071), will likely be useful at various points throughout the project as well.

My co-op experience has also contributed substantially to my preparation for this project. An add on feature to our project that we may implement if time allows after the main features have been implemented, is a feature to have different users that can log in and store assembly code in their profile. This would require working with a database, which I have had experience with in my first two co-ops as an Engineering Intern at ADVINT, and in my final co-op as a Technology Intern at CAS. In my third and fourth co-ops, which were with Crown Equipment Corporation as DevOps Engineering Co-op and an Embedded Development Co-op respectively, I learned a lot about style and project structure guidelines, which will be helpful for keeping our code organized, neat, and understandable. My Embedded Development Co-op involved working with low-level C code on embedded controllers, which gave me experience working with C. This will be a helpful understanding to have when working with C in our project. The most important co-op experience to this project is my final co-op with CAS. During this co-op, I worked heavily in TypeScript React, which is very similar to what the existing front end for this project is written in (JavaScript React). Having had a full summer to work with React and understand how to build an interface in that framework, will be very helpful in allowing me to hit the ground running with our project's user interface design.

I am excited about this project because it combines two parts of Computer Science that interest me. Learning about assembly code and how it interacts with computer architecture in Intro to Computer Systems is to date one of my favorite things I have learned about at UC. This project gives me the opportunity to spend more time learning about and working in that area, as well as to build a tool that can help pass that knowledge along to others. I will also need to learn more about how x86 machine code works specifically, since my original class on assembly code was taught using LC3, a simpler language. I am also excited for this project because it involves the design of a user interface, which I have found to be one of the more satisfying tasks in my Computer Science experience. It gives me the opportunity to design something that is both visually impressive and provides a quick indication of progress made, while also accomplishing something functionally complex, by interacting with backend services and performing complex data processing. It is also a tool that has potential to be very useful in the future, which provides more motivation than working on a project that is likely to not be touched after it is done.

For this project, we plan to follow a process that roughly corresponds to the agile method commonly employed in industry. We will take a few weeks at the beginning of the project to map out a good plan when it comes to large-scale tasks such as designing the backend compiler, the communication between the backend and frontend, and the changes that will need to be made to the frontend design. We will then start work on coding as soon as we reasonably can, so that we can make adjustments as necessary as we go. Throughout the project, we will decompose larger tasks into smaller more easily accomplished tasks and schedule goals to complete these tasks. We roughly plan to follow the industry standard two-week sprint format, of assigning tasks on one week, having a shorter check-in meeting the next week, and then having a longer review of completed tasks and assignment of new tasks the following week. We have established pretty

good design specifications with our advisor that we can use to confirm that we are successful.

The app must be able to take whatever C code is entered, compile it into assembly code based on the user's choice of assembly code types, and then that code must be able to be interpreted to power the animation of the computer's stack and registers.