

Virtual Development Environment in a Box

Dylan McDougall
Ian Orzel

Sponsor & Client

Dr. Stansifer

Meetings With Client

August 25th

August 31st

Motivation

Our motivation is to eliminate some of the difficulty that comes with setting up certain development tools that require exotic architectures, hardware, or software configurations for use in several classes at Florida Tech, such as *Compiler Theory*, *Operating Systems Concepts*, and *Programming Language Concepts*.

This creates many inconveniences for both students and faculty alike, as students have to figure out how to get the systems properly set up and faculty have to assist students with this setup process.

Goal

Our goal is to create a platform that provides a virtual environment for students to run software that requires specific system setups (i.e., hardware, architecture, software stack) that is convenient to the user.

This platform will be simple to use for both students and faculty and will support multiple architectures.

We want to at least have support for the Compiler Theory class completed before the semester ends so that our platform can be used and tested by students when the class is offered in the spring.

Features

- Easily download and run containers on your local machine of any platform loaded with environments for your classes to easily test and create programs for these courses, using an interface that was created by studying the users of the platform. These containers can have different architectures from the local machine. They will also contain different development tools, such as gcc and make, debugging tools, such as gdb, and specialized tools, such as linkers and assemblers.



Features

- Run your containers on any desktop platform, such as Windows, MacOS, and Linux.
- Import and export files to and from the container.
- Provide the ability to use the container's standard input and output.
- Use pre-built containers for the Compiler Theory, Operating System Concepts, and Programming Language Concepts classes. Also, create containers for new courses or for existing courses without having to create an entire virtual environment from scratch and share them with students.



Novelty

- The main novel feature of this system is that it will be created to be simple to use in an educational setting. The other tools that are similar to this are general-purpose and tied to specific platforms/software, which causes them to be more difficult to use than is needed for this audience.



Technical Challenges

- We plan to make our application cross-platform, and creating cross-platform applications is something we are not familiar with.
- For this project, we plan to use qemu to create and run our containers, but we are not familiar with this program. We need to learn how to create images in qemu and how to interact with these images in a programmatic setting.
- We plan to create virtual environments to be used for the Compiler Theory course, but we have not taken this course.



Milestone 1

- Compare and select technical tools for:
 - How the user interacts with the system
 - Emulating virtual environments
 - Distributing containers
- Provide small ("hello world") demo(s) to evaluate the tools for:
 - Containerization/virtualization
 - Container sharing
 - Using developer tools inside a virtual environment
- Resolve technical challenges:
 - Learn how to use qemu
 - Learn about the virtual environment that is needed for a Compiler Theory course
 - Learn how to set up an environment for different platforms

Milestone 2

- Implement, test, and demo a basic qemu image for use in Compiler Theory.
- Implement, test, and demo interaction with a qemu image through standard input and output.
- Implement, test, and demo file system interaction with a qemu image (allowing files to be easily transferred between user system and qemu system).

Milestone 3

- Implement, test, and demo a cross-platform utility to set up the environment to run our program for users.
- Implement, test, and demo an online platform to allow for the qemu images to be extracted from the internet and run on the local machine for the purpose of software development.
- Implement, test, and demo a pipeline that allows all of the implemented features to work together in order to be used in the Compiler Theory course next semester.

Questions?

Contact Information

Dylan McDougall (Student) - dmcdougall2019@my.fit.edu

Ian Orzel (Student) - iorzel2019@my.fit.edu

Ryan Stansifer (Faculty) - ryan@fit.edu

Thank You!