

Virtual Development Environment in a Box

Students: Dylan McDougall (dmcDougall2019@my.fit.edu)

and Ian Orzel (iorzel2019@my.fit.edu)

Faculty: Ryan Stansifer (ryan@fit.edu)

Client: Ryan Stansifer (ryan@fit.edu)

Progress of current Milestone:

Task	Completion	Ian	Dylan	Todo
Implement, test, and demo containers for Operating System Concepts and Programming Language Concepts	100%	0%	100%	
Test/demo of the entire system	100%	100%	0%	
Conduct evaluation and analyze results	100%	50%	50%	
Create user/developer manual	100%	90%	10%	
Create demo video	100%	0%	100%	

Discussion:

- **Implement, test, and demo containers for Operating System Concepts and Programming Language Concepts:** We made two containers using the container build tool finished in the previous milestones. The Programming Language Concepts container was trivial to create. It contains the standard Debian base with some added packages to provide tools for all the required programming languages. The Operating Systems Concepts container mainly involved configuring OS/161 inside the container, along with all the necessary tools.

- **Test/demo of the entire system:** After the test suite has been finished, it has been run to ensure that all of the features are working. The entire test suite has been run, and every test passes except for one, which is related to building a new container with a custom deb package.
- **Conduct evaluation and analyze results:** The test of the system turned up with almost all passes. The only test failure occurred during the process of building a new container with a custom Python package. We are currently investigating why this specific package fails to install during the test build.
- **Create user/developer manual:** A user manual has been created that features information needed for students and instructors to use our tool for coursework. This includes information about all of the commands available in the tool, examples of how to deal with common situations needed with the tool, and information on all of the components of the tool. It also features information about how to build new containers, how to create and interact with repositories, and the various pre-made containers that are available with the tool.
- **Create demo video:** We created a demo video which demonstrates the basic features of the jabberwocky tool. It demonstrates most of the student commands, and illustrates a possible use-case for the program by compiling the jabberwocky program for ARM inside of an ARM container.

Member Discussion:

- **Ian Orzel:** My focus for this milestone has been to get the test suite working and to create the user manual. First, I finished the test suite, which had not been fully completed

since the last milestone. After this, I have been running the test suite and making modifications to the tool to ensure that every test is working. Finally, I wrote the user manual.

- **Dylan McDougall:** I was responsible for creating the PLC and OSC containers, creating the demo video, and writing the portion of the user manual pertaining to the build process. I completed the container first after consulting with Dr. Stansifer and Dr. Ribeiro, and this was my main focus for the milestone.

Lessons Learned:

- Working on large projects in groups requires a lot of communication and teamwork. It is easy to become out of sync with group members, which causes problems while working on the project. Communication helps prevent later frustrations and ensure that the end product is functional and refined.
- It is important to ensure that progress is well-documented throughout the development of a piece of software rather than attempting to document it all at the end. The reasonings of various decisions get forgotten with time, and one may look back at previously written code (especially that of another person) and have no idea what its purpose is or how it works.
- Time management is the most important consideration when working on a project like this. While we were working on this, we were also enrolled in other classes and had other personal engagements. It was thus important to ensure that time was properly designated and distributed across all responsibilities. When we would mismanage time and not designate the proper time to this project, things became hectic and we began to sacrifice parts of the project and our process for the sake of time.

- Task prioritization is something we had to accustom ourselves with over the course of the project. There are a lot of things we wanted to implement but were only given a limited time frame to accomplish them. We had to decide what tasks needed to be accomplished now and which ones we would have to put off for later. For example, we made sure to implement the files command in Milestone 4 because it was a feature a lot of our users were requesting, and had to delay implementing the build command for Milestone 5.

Dates of Meeting with Client: (See Faculty Meeting Times)

Client Feedback for Milestone: (See Faculty Feedback)

Dates of Meeting with Faculty:

- 03/22
- 03/29
- 04/05
- 04/12

Faculty Feedback for Milestone:

- **Implement, test, and demo containers for Operating System Concepts and Programming Language Concepts:**

- **Test/demo of the entire system:**

- **Conduct evaluation and analyze results:**

- **Create user/developer manual:**

- **Create demo video:**

Faculty Advisor Signature: _____ Date: _____

Score for each member:

Ian	0	0.5	1	1.5	2	2.5	3	3.5	4	4.5	5	5.5	6	6.5	7	7.5	8	8.5	9	9.5	10
Dylan	0	0.5	1	1.5	2	2.5	3	3.5	4	4.5	5	5.5	6	6.5	7	7.5	8	8.5	9	9.5	10

Faculty Advisor Signature: _____ Date: _____