

## Virtual Development Environment in a Box

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

and Ian Orzel ([iorzel2019@my.fit.edu](mailto:iorzel2019@my.fit.edu))

Faculty: Ryan Stansifer ([ryan@fit.edu](mailto:ryan@fit.edu))

Client: Ryan Stansifer ([ryan@fit.edu](mailto:ryan@fit.edu))

Progress of current Milestone:

Task	Completion	Ian	Dylan	Todo
Implement, test, and demo the Container Creation wizard	0%	0%	100%	
Implement, test, and demo the ability to delete and rename containers	100%	100%	0%	
Implement, test, and demo Test Suite based on requirements	75%	75%	0%	Add more edge-case tests
Conduct evaluation and analyze results	50%	50%	25%	Wait for user feedback to come in
Create poster for Senior Design Showcase and eBook page	50%	50%	50%	

Discussion:

- **Implement, test, and demo the Container Creation wizard:** With the completion of this task, users of our software now have the ability to create their own customized containers with whatever software they want. Currently three architectures are supported: amd64, aarch64, and mipsel. I primarily considered two different tools when deciding how to go about this milestone: buildroot and debootstrap. Buildroot is a tool which allows for the creation of very minimal Linux systems with little to no user interaction, however, our requirements specify that the installation of Debian packages must be possible, and buildroot does not support this. Debootstrap is the tool I ultimately chose. It creates a Debian filesystem that can then be converted to a virtual hard disk image, for which I used something called guestfs-tools. The build process is executed by first running the new build-init command to create the necessary directory structure, then the build command to begin the build process. The build process occurs in these general steps:
  - User uses build-init to prepare a directory for creation (Milestone 4)
  - User uses the build command to begin the build process
  - The build process reads the manifest (Milestone 4) and decides what architecture to build for, packages to include, and scripts to execute.
  - The build process performs the initial first and second stages of the debootstrap.
  - The build process installs any packages specified by the user in the manifest and packages directory.

- The build process copies resources provided by the resources directory into the container's virtual home folder.
  - The build process executes any provided scripts
  - The build process converts the debootstrapped filesystem into a QEMU filesystem, extracts the Linux image, and generates a config file. Those components are then sent to a distributable container archive, which is by default compressed.
  - The build process exits.
- **Implement, test, and demo the ability to delete and rename containers:** These commands were relatively simple to construct and largely relied on the infrastructure that we have already set up. Stopped containers can now have all traces of them removed from the file system. Stopped containers can now also have their name changed, which is done by simply renaming the folder that they are stored in.
  - **Implement, test, and demo Test Suite based on requirements:** During this milestone, the framework for a test suite was built. To do this, a docker container was built so that the tests could run within it. Pytest was used for the tests themselves and was run in the container. In terms of tests themselves, there are around ten tests right now that test all of the basic functionalities covered by the requirements.

There were many issues that were encountered during the building of the test suite. Not only would it take a large amount of time to build and run the tests, but it would often run into issues that would then need to be addressed. Much of the time working on the test suite was spent waiting for the tests to run, looking at what went wrong, and trying to fix it. For instance, at first, the tests were not able to get any of the output sent by the CLI. After some experimentation, it was determined that this was due to a bug where output

was always getting sent to standard output instead of the output stream field of the CLI class. This had to be remedied in order to run the test suite. Another example was getting the container manager to be able to reach the repo server being hosted in another container during the tests. After a lot of testing, it was finally determined that we needed to find the ip address of the repo server while it was running and then send that to Pytest. All of these problems made this task significantly larger than was first expected and is the reason why this task did not get totally completed. Now that the framework of the test suite is up and running, adding more tests should be a much simpler process. Thus, all that's left to do is add more tests that check the edge-case functionality of the tool.

- **Conduct evaluation and analyze results:** For the first part of our analysis, we ran the test suite in its current state. Most of the tests passed except for two. The first one was a test to ensure that gcc was able to compile into SPARC binaries inside of the container for the Compiler Theory course. This test, for some reason, inconsistently works sometimes but not others. More experimentation will need to be done to determine why it fails on occasion. The other test that fails is the test of the upload container functionality. The cause of this is also currently not known, but there has not been extensive experimentation of the upload functionality. More of this will need to be done to determine why this does not work.

We are also waiting for the Compiler Theory students to fill out the survey we sent out to them a week ago. We will wait until the 23rd for results and then conduct an analysis on them. The surveys ask general questions about how are user are using the program and how satisfied they are with its current state, and we will use these results to make necessary improvements to our program.

- **Create poster for Senior Design Showcase and eBook page:** We have created the poster and ebook page that will be used to present our project during the Senior Design Showcase. These give a brief overview of our project and what it is for.

#### Member Discussion:

- **Ian Orzel:** The first thing that I did for this milestone was to create the rename and delete commands. This task proved very easy, and I completed it within a couple of days. After this was completed, my entire focus shifted to the test suite. As mentioned earlier, this task proved very time-consuming and required significantly more effort than was first expected. I spent a lot of time attempting to run the test suite, waiting for a failure to occur, and then attempting to fix the failure. Eventually, I was able to get the test suite working with some tests that ensured all of the requirements were fulfilled. I am still not done yet, as more tests are needed to check some of the edge cases. I will likely have to spend some time completing these during the next milestone. Also, after getting to this state of the test suite, I created the poster for the Senior Design Showcase.
- **Dylan McDougall:**  
I primarily spent my time working on the container creation tool, which is a crucial aspect of our project. Without it there isn't much point, as there would be no practical way for end users to create their own containers without an extensive knowledge of the internals of our program. With that done and the test suite nearing completion, we are entering the final stages of the project. One of our goals is to, by the end of Milestone 6, put out our first official stable release of the Jabberwocky tool, ie, version 1.0.0, and I think we are on track to achieve it.

Conducting the evaluation has proven difficult, however. Our users are unwilling to participate in our surveys because they don't care and probably have more pressing matters to occupy their time with. This is something that will need to be extended into Milestone 6 in order to get any meaningful results.

Next Milestone Matrix:

Task	Ian	Dylan
Implement, test, and demo containers for Operating System Concepts and Programming Language Concepts	10%	90%
Test/demo of the entire system	90%	10%
Conduct evaluation and analyze results	50%	50%
Create user/developer manual	75%	25%
Create demo video	25%	75%

Discussion of Planned Tasks for Next Milestone:

- **Implement, test, and demo containers for Operating System Concepts and Programming Language Concepts:** Two other great examples of courses that would benefit from having containers are Operating System Concepts and Programming Language Concepts. Operating System Concepts requires a custom lightweight operating system to be running so that students can build it. Programming Language Concepts requires students to compile and run programs written in four different programming languages. We would thus like to create containers for these two courses.
- **Test/demo of the entire system:** We would like to get the test suite into a complete state and have our tool pass every test 100% of the time. The first step to doing this is to

ensure that some extra tests are added so that the test suite can be in a complete state.

After this, we have to identify test failures, determine why they are happening, and fix the bugs in the tool that are causing them.

- **Conduct evaluation and analyze results:**
- **Create user/developer manual:** We would like to create a user manual to be read by students and faculty for them to learn more about what our tool is and how to use it. This should assist with them using it with their coursework.
- **Create demo video:** We would like to create a video of our tool in action to further assist students and faculty with learning our tool. Learning from watching others do can be very helpful for learning new things, especially when reading a manual can feel like a daunting task.

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

Client Feedback for Milestone: (See Faculty Feedback)

Dates of Meeting with Faculty:

- February 15
- February 22
- March 1
- March 8

Faculty Feedback for Milestone:

- **Implement, test, and demo the Container Creation wizard:**

- **Implement, test, and demo the ability to delete and rename containers:**
- **Implement, test, and demo Test Suite based on requirements:**
- **Conduct evaluation and analyze results:**
- **Create poster for Senior Design Showcase:**

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: \_\_\_\_\_