

Milestone 4

Task Matrix

Task	Completion	Ian	Dylan	Todo
Implement, test, and demo specifications for container creation	100%	0%	100%	
Implement, test, and demo container repositories	90%	90%	0%	Testing
Implement, test, and demo more intuitive file system interaction	100%	5%	95%	
Implement, test, and demo fixes found by students in the Compiler Theory course	100%	60%	40%	



0

Compiler Theory Bug Fixes

Updating

- Implemented new update command
 - Finds newest release on GitHub
 - Downloads and installs if different from current version



Teaching

- We had to teach students how to use our tool
 - Held a seminar during class time
 - Held an evening workshop



Bug Fixes

- Absolute versus relative paths
- The '~' path
- Assuming file names
- Informing user of errors
- Fix Ctrl+C
- Fix start
- Fix server-halt ungraceful exit

1

File System Interaction

Changes to get-file, send-file

• `send-file [container] [source] [dest]`

- The destination argument is handled differently now.
- If [dest] is a directory, the destination file will inherit the name of the source file.
- If [dest] is not give, the destination file will inherit the name of the source file and will be copied to \$HOME.



File viewer

- Provides a graphical file manager used to view a container's virtual file system.
- Uses an embedded FileZilla instance that connects to the virtual file system over SFTP.



Demonstration

- <https://www.youtube.com/watch?v=XyD3aYQjn3g&list=PLRkjHslsoL3nRKXz9KblvxJtpSagLtrpK>

2

Repositories

Repositories

- Get a list of archives
- Download a particular archive
- Upload your own archive (after providing “authentication”)



Container Manager

- Allow users to add/update repositories
- Allow users to download archives
- Allow users to archive their own containers
- Allow users to upload archives to repositories



Download Example

- <https://youtu.be/4aoL4kZckmE>

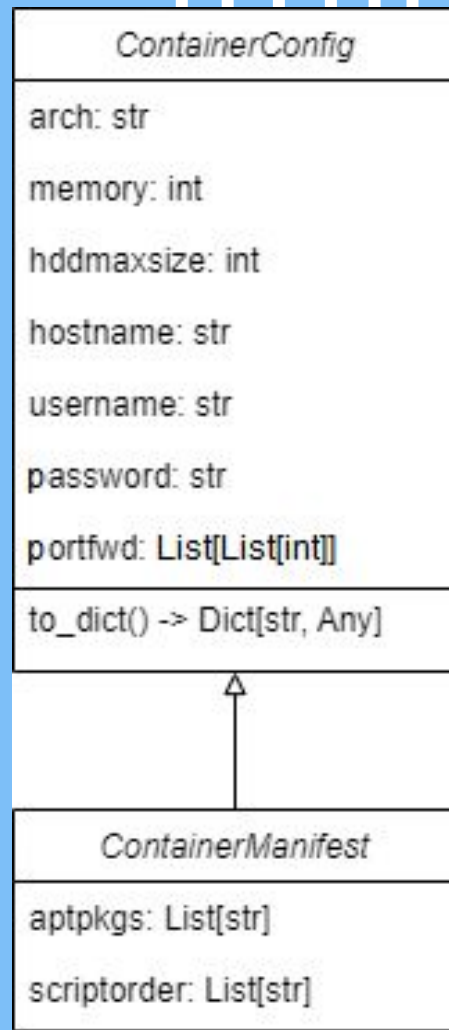


3

Container Creation Specification

Container Definition

- **Req. 22** Users can specify the architecture that their container will use.
- **Req. 23** Users can specify whether to include the standard development tools in Linux when they build a new container. This includes gcc and gdb.



Directory Structure

- **Req. 24** Users can provide [Debian] packages to be included and installed into their containers.

```
✓ build
  > dist
  > temp
  > packages
  > resources
  > scripts
  {} manifest.json
```

```
{
  "manifest": 0,
  "arch": "x86_64",
  "memory": 500,
  "hddmaxsize": 10,
  "hostname": "debian",
  "portfwd": [],
  "aptpkgs": [],
  "scriptorder": []
}
```



4

Milestone 5 Goals

Container Creation

- In this goal, we will actualize the work done in the Container Definition goal of the previous milestone in order to fulfill the Container Creation requirements.
- **Req. 21** Users can create new containers using a special command that will come built with various options... it will create a new container that can be utilized like other containers.



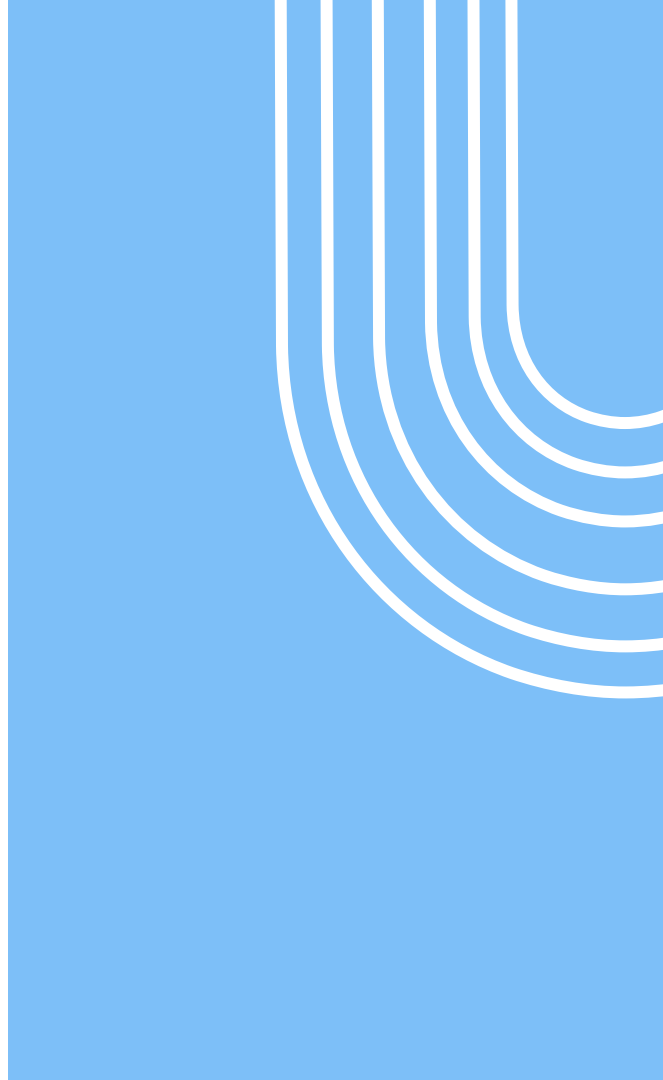
Extra Commands

- Delete Command
 - Delete a container from the file system
- Rename Command
 - Change how users refer to a container



Test Suite

- Create framework for testing project
 - Difficult to connect all parts
- Create tests for all of the requirements



Evaluation and Analysis

- Check results of running test suite
- Survey Compiler Theory students
- Run experiments on random students



Showcase Poster

- Senior Design Showcase requires a poster
- We need the poster to summarize our project



A dark, menacing creature with red eyes and sharp teeth is shown in a dark, misty forest. The creature has a grotesque, multi-limbed appearance with long, thin appendages. The background is a dense, dark forest with gnarled trees and a hazy, blue-tinted atmosphere.

Thank You

Questions?