Virtual Development Environment in a Box

Developers & Clients

Students:

Ian Orzel

Dylan Mcdougall

Faculty/Client:

Ryan Stansifer

- Met on 01/11

0

The Big Picture

Goal and Motivation

Goal

- Provide virtual environments with different system
- Easy to use for students/faculty
- Support for Compiler Theory course

Motivation

- Setting up environments for coursework is tricky
- Saves time for students/faculty
- Prevents “works on my machine” issues

Approach

- Use on-demand virtual development environments custom-tailored to your courses.
- Use the pre-made environments created for the Compiler Theory, Programming Languages, and Operating Systems courses.
- Easily emulate foreign guest architectures other than that of your host machine without even having to think about it.
- Be able to create containers with whatever custom software you need to distribute to your students for a specific course.

Algorithms and Tools

- QEMU Emulator
- PyInstaller
- UNIX TCP Sockets
- SSH

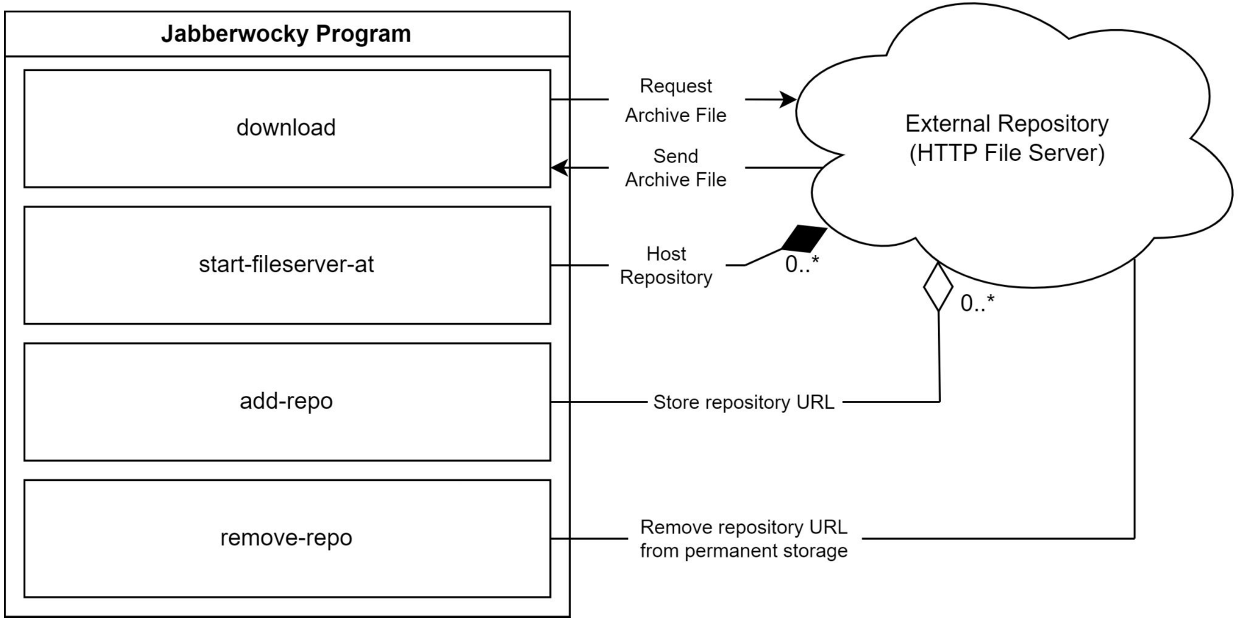
Novel Features

- We have created this project with an educational audience in mind, and are custom-tailoring it for specific classes at Florida Tech.
- Compared to andrew.cs.fit.edu, our software runs entirely locally.

Technical Challenges

- Making software improvements while it is in use
- Using Qemu to automate the container-creation process
- Servers need to be hosted for repositories

Design


Jabberwocky Program
download
start-fileserver-at
add-repo
remove-repo
Request Archive File
Send Archive File
Host Repository
0..*
0..*
Store repository URL
Remove repository URL from permanent storage
External Repository (HTTP File Server)

Evaluation

- Survey for Compiler Theory students (rate 1-5):
  - Please rate how easy-to-use this tool is
  - Please rate how likely you would be to use this tool instead of setting up environments on your personal computer
  - Please rate how easy it is to interact with a container's file system
  - Please rate how easy it is to run commands on a container
  - Please rate how easy it is to manage the containers installed on your machine (starting/stopping and installing)

Evaluation

- Time length it takes to complete tasks
  - Create a new container with given specifications
  - Install a given container and look into its file system
  - Add source code files into a container, compile them, and then run them
- Implement test suite

1

Second Semester

Progress Summary

Module/Feature	Completion %	To do
Compiler Theory Container	95%	Work with students to improve
Operating Systems Container	0%	Deduce requirements for this container Create image that fulfills requirements
Programming Languages Container	10%	Create image that includes all needed compilers
Container Manager	80%	Add delete command Implement download, archive, add-repo, update-repo Fix bugs that occur during edge cases
Container Storage	0%	Create code for container-hosting repositories Ensure container manager can communicate with repositories
Container Creation	0%	Create wizard that converts user inputs to specifications for a container Allow containers to be created to those specifications

Milestone 4

- Implement, test, and demo specifications for container creation
- Implement, test, and demo container repositories
- Implement, test, and demo more intuitive file system interaction
- Implement, test, and demo fixes found by students in the Compiler Theory course

Milestone 5

- Implement, test, and demothe 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

Milestone 6

- 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

2

Milestone 4

Container Creation Specifications

- “Container Creation wizard” by Milestone 5
- We will decide how the end user will create a container, what options they can choose, which ones they can’t, how they choose them, etc.

Repositories

- Users must be able to upload and download containers
- Server code must be created for storing containers, called repositories
- Container manager must be able to interact with repositories

File System Interactions

- Modify the virtual filesystem of the container in a more intuitive way.
- send-file and get-file commands are sufficient to meet our technical requirements, but may become tedious.

Compiler Theory Bug Fixes

- Compiler Theory course is using our tool this semester
- Work with them to identify bugs and possible improvements
- Implement changes and get feedback from them

THANKS!

Any questions?