***Computer Science***

| **Project Name** | **Jabberwocky** |
| --- | --- |
| Team Member(s): | Ian Orzel and Dylan McDougall |
| Faculty Advisor(s): | Dr. Ryan Stansifer |

Project Description:

The purpose of the Jabberwocky project is to create a cross-platform solution for the creation and distribution of virtual software environments primarily intended for software development in an educational setting. These virtual environments may emulate foreign architectures and contain different development tools,

such as gcc and make, debugging tools, such as gdb, and specialized tools, such as linkers and assemblers.

Features:

Jabberwocky containers are preloaded with custom software and development tools, which can be trivially installed onto a given host machine. These containers can then be interfaced with by running their internal commands individually, by entering a virtual shell session within the containers, and by interacting with their virtual filesystems, either through a command-line or graphical interface. The Jabberwocky tool also provides its users with the ability to create their own custom containers. The user may choose an architecture to emulate, choose software packages to include in the container, choose any resources to include in the container, execute scripts inside the container for system configuration purposes, and forward any necessary ports from the container’s virtual networking environment to the host machine. These containers can then be distributed either as single compressed archive files, or via a Jabberwocky container repository.

Technical Challenges:

Some of the major technical challenges we encountered during the course this project were finding an effective way of emulating foreign architectures, finding the best way to distribute containers, ensuring the program is cross-platform. We solved the emulation question though the use of minimal QEMU virtual machines. The files associated with this virtual machine, as well as a configuration file instructing the main tool as to how to execute this virtual machine, are distributed as tar archives. These archives can either be distributed individually or consolidated in a repository. We did our best to only depend on cross-platform tools, and our program is packaged and distributed using a Python module called PyInstaller.

Evaluation:

Near the beginning of the Spring 2023 semester, a development release of the Jabberwocky tool was shared with the students of the Compiler Theory class. We are continually checking in on these students to gauge the quality of our tool. So far, they have had success in using it for class projects. Otherwise, our project has a test suite which we will use to ensure that all of its components are functional. The way we will evaluate the success of our project is by evaluating user feedback and the success of our tests.

Future Work:

We anticipate that our project will need minimal maintenance after the first stable release. Some things that might need to be done are to provide help to professors and students wishing to use our tool in the future, and to ensure it remains working with future changes that might occur to tools our project depends on, such as QEMU. Also, future bug fixes may be for currently unknown bugs in the code. We may also need to advise professors in creating their own containers for their courses.