

TOUCHE: Coding Competition Software

Tyler Garcia

Department of Computer Science and Engineering, Taylor University, Upland, Indiana 46989

Introduction

Touche is a coding competition software that is used to make the entire experience of running a competition or competing in a competition as seamless as possible. In order to do this, the system will set up a database, give a easy to use interface, and automate as much as possible. Setting up Touche was something that was difficult to setup on new machines as there was a lot of things set to a default server address which means that I spent a lot of time making new documentation and learning how to best setup the server such that it will hopefully work on a variety of Unix distributions This took up a majority of the first week or so of the month and from there I, and my team, ran into several bugs which had to be fixed.

Apache

Setting up Apache as a web server was the first thing that had to be done to get this project started up. The initial Apache setup was easy until I got around to the PHP part. Having never done this sort of thing, getting PHP up and running turned out the be a long process where I continually missed small things and would have to start over again. Setting up Apache to use user directories was a must for this project so that the team could work on different aspects of the code without messing each others code up so I had to learn how to do this as well. Eventually, after a lot of mess ups Apache was finally set up and running, not to mention I could finally setup the entire server again from memory even if it did break.



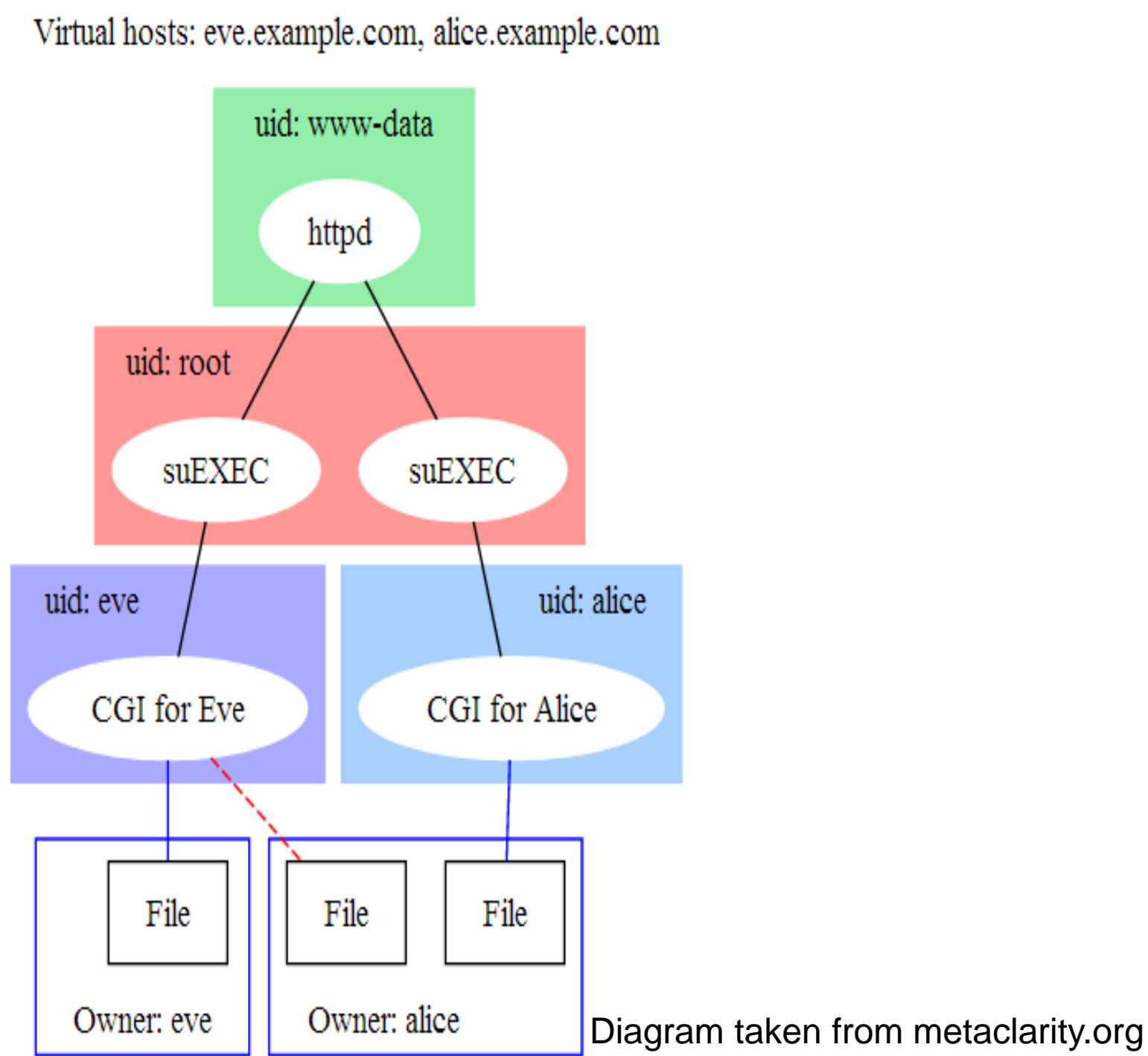
MySQL

MySQL is a pretty straightforward package to install so there wasn't much in the way of setup I had to do for this. In terms of Touche, I had to modify the database creation SQL file to work for the newer version of MySQL since the file was from 2005 which means it had outdated syntax. I also fixed some hiccups in the database that caused errors with judging problems incorrectly.



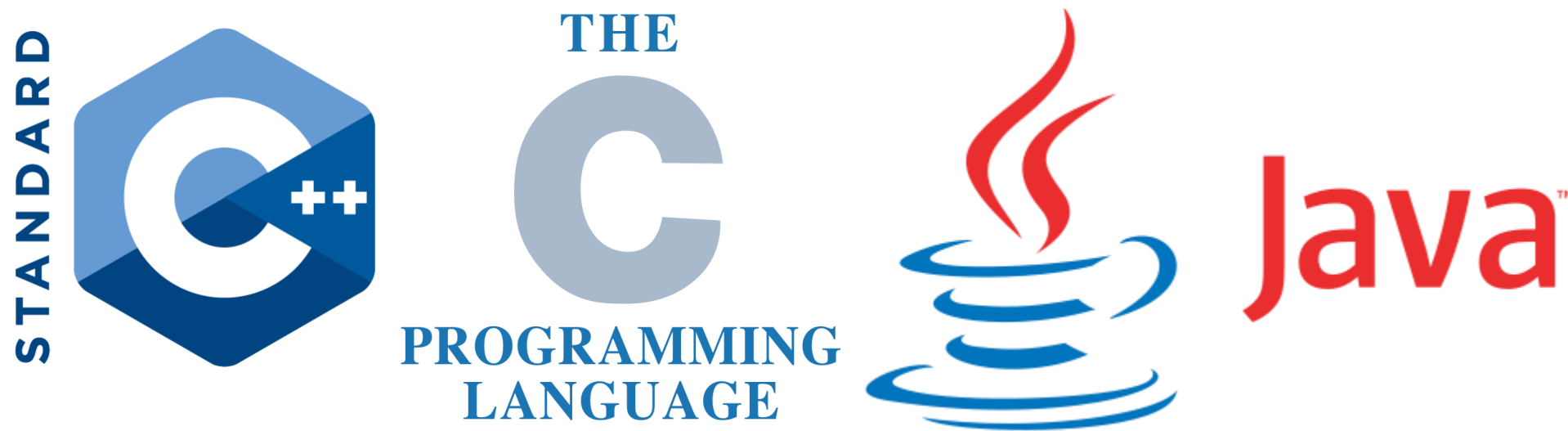
SuEXEC

SuEXEC is a wonderful tool that allows PHP scripts to run as the user from whose directory the file is being accessed from. Setting this tool up, as great as it is, turned out to be a huge headache but after roughly three days of working at, with help, I was able to get it setup and running correctly. Below is a diagram showing how suEXEC works in an easy to understand way.



Jails

Jails are locations in the filesystem that will be used as a home directory for the server to use when the chroot system call is called. Setting up the C++ and C jails was pretty easy as I just had to copy over the compilers and some libraries needed. Java, on the other hand, turned out to be a lot harder. After spending a few days trying to copy just the compiler and files needed for the compiler to run I came to the realization, with help from Dr. Geisler, that I was going to have to copy the entire Java JDK directory into the jail. After doing this all the jails were functioning correctly and we were able to run full test competitions.



Challenges

Since I was the person that was doing server stuff that none of us had seen in detail, I had quite a few challenges. A lot of these challenges seemed to stem from my messing up either the suEXEC or the PHP files. That being said, I also ran into some walls when it came to setting up jails. I would get the jails to work using chroot from the command line but not when using the software. Of course, turns out this stemmed from suEXEC not working correctly. Another challenge I had was the challenge of setting up the user directories to have the correct permissions. Of course, after we figured out the issue with permissions me and the team was able to fix the issue whenever it arose.



Future Improvements

- Here is a list of improvements that I had planned on adding to the software but did not have time to either complete or didn't have the ability to do.
- Python Support
- Java setup without need of the entire JDK
- Test compile for teams
- These three improvements would make the software much more appealing to anyone that is looking for a coding competition software package.

Conclusions

I am happy with what Touche has developed into after a month of work. All of the compilers are now up to date, the design is updated, the server documentation is readable, and everything is a lot more user-friendly. I think that with more time, Touche could turn into something even more amazing than it is now. Working on fixing other people's code has helped me keep in mind the fact that when I write commercial code that someone in the future will have to eventually look at my code and understand what is going on.

