

Programming Leftovers

By *Roy Schestowitz*

Created 21/10/2021 - 10:51pm

Submitted by Roy Schestowitz on Thursday 21st of October 2021 10:51:39 PM Filed under [Development](#) [1]

- [How C++ Modify Arrays in Function](#) [2]

Arrays have been widely known among programmers and developers. We have been using arrays in almost every structural language to object-oriented language. As we know, arrays store more than one value in their indexes, and we also modify the arrays. So, in today's article, we will be deliberating how to modify the arrays in functions of C++. Start by logging in from the Linux system and launching the terminal with the ?Ctrl+Alt+T? shortcut.

- [Digging into Julia's package system \[LWN.net\]](#) [3]

We recently looked at some of the changes and new features arriving with the upcoming version 1.7 release of the Julia programming language. The package system provided by the language makes it easier to explore new language versions, while still preserving multiple versions of various parts of the ecosystem. This flexible system takes care of dependency management, both for writing exploratory code in the REPL and for developing projects or libraries.

[...]

For a while I've thought that the Plots package needed a particular feature. This morning I cloned the project to my computer, added the feature, made a pull request on GitHub, made a change suggested by one of the maintainers, and got it approved. The entire elapsed time for this process was about five hours. In this section I'll describe two more package system commands that make it easier to hack on public packages.

In the REPL, I entered package mode, then executed `develop Plots`. This command, which can be shortened to `dev`, clones the named package's Git repository to the user's machine in the directory `.julia/dev/`. Since `Plots` is a big package with many source files, this took about two minutes.

This command also alters the environment so that using `Plots` imports from the version under development, rather than the official version. The command `free Plots` returns to using the official version. One can switch back and forth between these two incarnations of the package freely, as subsequent `dev` commands won't download anything, but simply switch back to the development version.

I entered the development directory and created a branch for my feature with the git checkout `?b` command. The package manager doesn't require this; it's happy to let you mangle the master branch. But I had plans to ask that my feature be merged into master, and needed to create a branch for it. Packages under `develop` are loaded from the file tree, not from the Git repository.

Then I wanted to edit the function to add my feature. But where is it? `Plots` has 37 files in its `src` tree. Because of multiple dispatch, each function can have dozens of methods associated with it, all with the same name. This makes finding a particular method in the source difficult to accomplish with simple `grep` commands.

- [A QEMU case study in grappling with software complexity \[LWN.net\]](#) [4]

There are many barriers to producing software that is reliable and maintainable over the long term. One of those is software complexity. At the recently concluded 2021 KVM Forum, Paolo Bonzini explored this topic, using QEMU, the open source emulator and virtualizer, as a case study. Drawing on his experience as a maintainer of several QEMU subsystems, he made some concrete suggestions on how to defend against undesirable complexity. Bonzini used QEMU as a running example throughout the talk, hoping to make it easier for future contributors to modify QEMU. However, the lessons he shared are equally applicable to many other projects.

Why is software complexity even a problem? For one, unsurprisingly, it leads to bugs of all kinds, including security flaws. Code review becomes harder for complex software; it also makes contributing to and maintaining the project more painful. Obviously, none of these are desirable.

The question that Bonzini aimed to answer is "to what extent can we eliminate complexity?"; to do that he started by distinguishing between "essential" and "accidental" complexity. The notion of these two types of complexity originates from the classic 1987 Fred Brooks paper, "No Silver Bullet". Brooks himself is looking back to Aristotle's notion of essence and accident.

Essential complexity, as Bonzini put it, is "a property of the problem that a software program

is trying to solve". Accidental complexity, instead, is "a property of the program that is solving the problem at hand" (i.e. the difficulties are not inherent to the problem being solved). To explain the concepts further, he identified the problems that QEMU is solving, which constitute the essential complexity of QEMU.

- [Notes from the Git Contributors' Summit 2021, virtual, Oct 19/20](#) [5]

we held our second all-virtual Summit over the past two days. It was the traditional unconference style meeting, with topics being proposed and voted on right before the introduction round. It was really good to see the human faces behind those email addresses.

32 contributors participated, and we spanned the timezones from PST to IST. To make that possible, the event took place on two days, from 1500-1900 UTC, which meant that the attendees from the US West coast had to get up really early, while it was past midnight in India at the end. I would like to thank all participants for accommodating the time, and in particular for creating such a friendly, collaborative atmosphere. A particular shout-out to Jonathan Nieder, Emily Shaffer and Derrick Stolee for taking notes. I am going to send out these notes in per-topic subthreads, replying to this mail.

- [Notes from the 2021 Git Contributors' Summit](#) [6]

For those who are curious about where the development of Git is headed: Johannes Schindelin has posted an extensive set of notes from the just-concluded Git Contributors' Summit.

- [How to find a substring in Python](#) [7]

Python is a versatile language having many built in methods and libraries. Strings and substrings are an important part of every programming language; python provides different methods to deal with strings and substrings, we check if a python string has a substring for a variety of reasons, but conditional statements are the most typical application. To find substrings in a string, python language provides many predefined methods.

- [How to find the average of a list in Python](#) [8]

Average (Arithmetic mean) is a mathematical function which is calculated by adding the

numeric values in the list and dividing them by the count of numbers of the list. Python provides several built-in mathematical functions; consequently it provides different ways to calculate the average of a list.

- [Ian Jackson: Going to work for the Tor Project](#) [9]

I have accepted a job with the Tor Project.

I joined XenSource to work on Xen in late 2007, as XenSource was being acquired by Citrix. So I have been at Citrix for about 14 years. I have really enjoyed working on Xen. There have been a variety of great people. I'm very proud of some of the things we built and achieved. I'm particularly proud of being part of a community that has provided the space for some of my excellent colleagues to really grow.

- [This Week In Rust: This Week in Rust 413](#) [10]

- [Oracle Releases GraalVM 21.3 With Java 17 Support, Other Enhancements](#) [11]

Oracle has published its latest quarterly update to GraalVM, the open-source Java JVM/JDK implemented in Java that also supports other execution modes and programming languages from Python to R to Ruby.

Given last month's release of Java 17 / OpenJDK 17, GraalVM 21.3 has added Java 17 support. Plus there are many other improvements to its various language front-ends and other components. Some of the GraalVM 21.3 highlights include:

- Java 17 support with GraalVM builds based on Oracle Java 17 and OpenJDK 17. OpenJDK 11 also continues to be supported while OpenJDK 8 is no longer supported by GraalVM.

- [Security updates for Thursday](#) [12]

Security updates have been issued by Debian (python-babel, squashfs-tools, and uwsgi), Fedora (gfbgraph and rust-coreos-installer), Mageia (aom, libslirp, redis, and vim), openSUSE (fetchmail, go1.16, go1.17, mbedtls, ncurses, python, squid, and ssh-audit), Red Hat (java-1.8.0-openjdk and java-11-openjdk), Scientific Linux (java-1.8.0-openjdk and java-11-openjdk), SUSE (fetchmail, git, go1.16, go1.17, ncurses, postgresql10, python, python36, and squid), and Ubuntu (linux, linux-aws, linux-aws-hwe, linux-azure, linux-azure-4.15, linux-

dell300x, linux-gcp, linux-gcp-4.15, linux-hwe, linux-kvm, linux-raspi2, linux-snapdragon, linux, linux-bluefield, linux-gcp-5.4, linux-hwe-5.4, linux-kvm, linux-oem-5.10, and linux-oem-5.13).

Development

Source URL: <http://www.tuxmachines.org/node/157061>

Links:

- [1] <http://www.tuxmachines.org/taxonomy/term/145>
- [2] <https://linuxhint.com/cpp-modify-arrays-in-function/>
- [3] <https://lwn.net/Articles/871490/>
- [4] <https://lwn.net/Articles/872321/>
- [5] <https://lwn.net/ml/git/nycvar.QRO.7.76.6.2110211129130.56@tvgsbejvaqbjf.bet/>
- [6] <https://lwn.net/Articles/873629/rss>
- [7] <https://linuxhint.com/find-substring-python/>
- [8] <https://linuxhint.com/find-average-list-python/>
- [9] <https://diziet.dreamwidth.org/10255.html>
- [10] <https://this-week-in-rust.org/blog/2021/10/20/this-week-in-rust-413/>
- [11] https://www.phoronix.com/scan.php?page=news_item&px=GraalVM-21.3-Released
- [12] <https://lwn.net/Articles/873601/rss>