Fortran Package Manager's First Birthday

Brad Richardson

It was about 1 year ago that I attended my first Fortran standards committee meeting, more formally known as a J3 committee meeting. It was there that I met some brilliant people I'm now happy to call colleagues. It was a very interesting experience, and I'd recommend it to anyone interested in learning about how design by committee languages in general, or Fortran specifically are developed. But what was most exciting was that by the end of the meeting, Ondrej and I had decided to collaborate on the prototype of the Fortran Package Manager (fpm).

Ondrej had already been toying around with the start of an implementation written in Rust. He had chosen Rust because he wanted to model the Fortran package manager after Rust's, named Cargo. It was an idea I'd had rolling around my head for a while as well. I suggested to him that we switch to using Haskell, as I already had a quite complete Fortran build system written in Haskell using the Shake library. He agreed, we put together a rough outline of the features we'd need and some conventions for packages, and set to work on a prototype.

The meeting was the last week of February 2020. I show from the git logs for fpm, that a version enabling dependencies was merged at the end of May. Logs from my own projects show that I was switching them over to using fpm in late May to early June. It was not long afterwards that others started to try it out and suggest features and how the user interface should be designed.

It was very quickly that many interested in using fpm also expressed some interest in helping with its development. There was one problem. Most people were not familiar with Haskell, making that a significant barrier to their being able to contribute. But everybody had significant experience (or at least interest) in Fortran. It was thus that we decided we would rewrite fpm in Fortran, enabling a larger portion of its intended users to also be contributors. With the initial prototype usable for developing libraries, and many already converting their existing libraries to be fpm compatible, it didn't seem quite so daunting a task as it otherwise would have. On July 21st, the start of that effort was submitted and merged.

It turned out to be a great idea. By mid November we created an initial Alpha (version 0.1.0) release. It included both the Haskell and Fortran versions, but already they were nearly equivalent in terms of capabilities. And while I did have an early hand in contributing to the Fortran version, a large fraction of the work was done by contributors who weren't even involved in the project until soon before or even after the decision to rewrite in Fortran. I must express a very hearty kudos and thanks to those contributors.

At this point I don't think I've made source code contributions in more than a couple of months. I stay active and engaged in the discussions and make suggestions for the design of the user interface. When I first started the project I had some concern that I might end up being BDFL, and if I ever got busy with other things the project might languish and die. It's clear now that the community isn't going to let that happen. They probably don't really even need me anymore.

To have gotten to this point in less than a year is an incredible success story for an open source project. I attribute much of the success to the fortran-lang.org project for helping to publicize the project and attract users and contributors. Ondrej and Milan have already written about their experiences over the last year working on that project. But I also believe it underscores just how desperately the Fortran community has needed investment in modern tools.

I think it's clear now that the open source Fortran community is fully on board with developing and adopting modern tools to make their lives easier. They're also committed to making the language more inviting to new users, including having a curated and well written collection of tutorials in a convenient, centralized place. There is still plenty left to do, but I can see now that the goal I had in mind not much more than a year ago is well on its way to being accomplished, if not already there.

1 References