Community & Contributing

CIS 198 Lecture 14

Contributing to Rust

- For the final project (or for fun!) you're allowed to contribute to the Rust language project, or to several other Rust projects.
- Read the contributing guide here to get started.
- Also check out the issue tracker.
  - There are 2,288 issues open as of time of writing!
  - Filter by E-Easy, E-Medium, or E-Mentor to find good starting issues.

Contributing to Rust

- There are three(ish) main parts of the Rust project:
    - `rustc`, the Rust compiler
    - `libstd`, the Rust standard library
    - documentation
- If you want to contribute to core `rustc`, beware that you may find it very difficult!
- There's no real distinction between `rustc` & `libstd` in the issue tracker, but it's usually pretty obvious what an issue is about.
- If you wish to contribute to Rust for the final project, don't *only* work on documentation.

Etiquette

- If you wish to contribute, make sure you also read and follow Rust's Code of Conduct.
- Be polite. Be organized. Don't create unnecessary work for other people. Test your code well. Use GitHub properly (issues and pull requests).

Fork-Pull Request Model

- When you want to work on a feature, start by forking the Rust repo on Github.
- When you've completed whatever feature you're working on, make a pull request from your fork against the main Rust repo, and await review.
- You'll be assigned a reviewer by bors, the Rust bot.
- Your reviewer may ask you to make changes, provide more tests, etc. to your code.
- Once review is complete, your changes will be "rolled up" by bors into a changeset and merged into `master` when appropriate.

RFCs

- If you want to propose a *larger* feature for Rust, Cargo, or Crates.io, you should create an *RFC*, or Request For Comments.
    - e.g. semantic/syntactic changes, additions to `std`, removing language features...
- Before you submit an RFC, you may want to discuss it on the Rust internals forum, IRC, etc. and talk to the core team about it first.
    - This is suggested to help you flesh out ideas & make sure an RFC won't get rejected outright.
- If you think your idea is still worth proposing, write a formal RFC using the template on the RFCs repo, following the process here.

Resources

- Contributing guide
- Issue tracker
- /r/rust on Reddit
- #rust on IRC
- Rust Internals Forum

Rust Community Projects

- Servo
- Piston
- Redox
- Others

Servo

- A parallel browser engine project developed in Rust by Mozilla.
- Contributing guide.
- Quick-Start guide.

Piston

- A game engine under development in Rust.
- Contributing
- Quick-Start

Redox

- A brand-new operating system written in Rust!
- Contributing

Other Projects

- Look at libs.rs, crates.fyi and Awesome Rust for more ideas.