Rustに入門してみた

---

agenda

1. What's Rust?
2. I think for Rust.
3. I'm Interesting to Rust! why?
4. getting started. - to "Hello, World" -
5. FizzBuzz with test
6. Summary

What's Rust?

ググってみました。

Wikipediaさん

Rust言語は速度、並行性、安全性を言語仕様として保証するC言語、C++に代わるシステムプログラミングに適したプログラミング言語

Rustはマルチパラダイムプログラミング言語であり、手続き型プログラミング、オブジェクト指向プログラミング、関数型プログラミングなどの実装手法をサポート

基本的な制御フローはC言語に似ている

実行時速度性能はC言語と同等程度

What's Rust?

インタネの海におけるRustの話題（偏りあり）

所有権
むずい
わからん
所有権
ライフタイム
所有権
所有権

I think for Rust.

ちょっとかじった感じの目黒的Rust解釈

Rust = (C + OOP / 2) * safety^2 * Modern

つまり、C言語にOOPをちょっと混ぜて、安全性にめっちゃ配慮して、今風のエッセンスを仕込んで仕上げた感じ。

I think for Rust.

所有権とライフタイムという仕組みによるメモリ管理

変数の所有権（どこで更新するか）とライフタイム（いつまで保持するか）を明確にコーディングすることでコンパイル時に変数の扱いをチェックする。

実行時にはライフタイムに従って自動で破棄。これにより、メモリリークと、いつの間にかメモリが書き換わってるという調査困難なバグを生み出さない。
GCではないのでメモリ破棄のタイミングをコントロールできる。重くない。

C言語では、readonlyな引数はconstをつけることで明示できたが、わざわざreadonlyを書いて更新できなくするという方法は面倒ではやらなかった（と思う）。
Rustでは、引数はデフォルトreadonlyで、更新したいなら面倒な手続きを踏めということになった模様（たぶん）。

I think for Rust.

エラー処理

例外は使わない。どこまで例外を投げるか、どこでどの例外を処理するか、という面倒なことに頭を使わない。
~~返り血~~ 返り値で異常を判断するシステム。

テスト

テストフレームワークなどいらん。言語仕様として単体テストが書ける。

I think for Rust.

OOP

オブジェクト指向が**ちょっと**

Rustにはclassはない。構造体に関数をimpl(ements)することでオブジェクト指向を表現する。
継承はない、代わりにトレイトという仕組みを使うことができる（が完全な代用ではない）。

一般的なOOPの言語とは違う思想でオブジェクト指向が組み込まれている。

I think for Rust.

ポインタは健在

Cといえばポインタ。
そのポインタの考え方は健在だし、書き方も&をつけるという形でCのまま。

I'm Interesting to Rust! why?

ググった感じと、実際に触った感じからCの正統系進化という感想。（対照的に言うならC++は闇堕ち系進化）
そこがC歴15年のエンジニアには良く響いた。
トレンド？そんなんしるか

所有権・ライフタイムの仕組みはCの弱点を上手くフォローしてると思う。

総じて、社内に多くいるC Programmerに広めていきたいと思える。

I'm Interesting to Rust! why?

チュートリアルという学習サイトがある。
日本語版がある。
なかなか内容が分厚く、これ読むだけでRustをだいたい抑えられそう
https://doc.rust-jp.rs/book/second-edition/

getting started. - to "Hello, World" -

1. 以下からインストーラをダウンロード
   https://win.rustup.rs/
2. インストーラでインストール（コマンドプロンプトが起動される）
3. コマンドラインから以下を実行

```
cargo new hello_world --bin
cd hello_world
cargo run
-> Hello, World!
```

FizzBuzz with test

https://github.com/m-megmog-m/fizzbuzz

Summary

C言語は、今世の中で使われているすべての言語に影響を与えているといって差し支えないAmazingな言語であるが、より便利で簡単な言語が多数生まれている中では、もう現役でいるべきではない。

その後継として、今まではちょっと不良な感じのC++がブイブイきてたが、ここにきてようやく優等生な感じのRustが登場した。
この優等生クンは、師匠のカタは守りつつ師匠のイマイチなところは克服し、さらに他流派の良いとこも取り込もうとする熱心な子であります。
この流派はちょっと堅苦しく回りくどいところはあるけど、それも世の中の動きに合わせて柔軟に変わる可能性を秘めている。
良さはそのままに進化するC系言語として注目に値すると思う。