.Deb

銀河系唯一のDebian専門誌

2020 年 10 月 17 日

rust を使う

デビアン勉強会

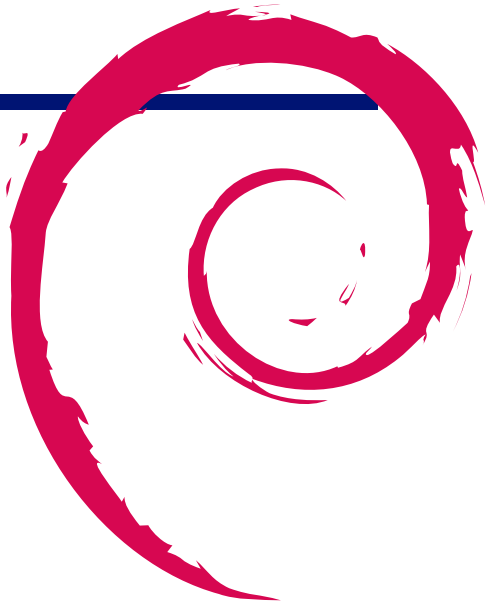
目次

1 最近の Debian 関連のミーティング報告 2
1.1 2020 年 9 月度 東京エリア・関西合同 Debian 勉強会 . . . 2

2 事前課題 3
2.1 dictoss . . . 3
2.2 yy_y_ja_jp . . . 3
2.3 NOKUBI Takatsugu (knok) 3
2.4 Hiroyuki Yamamoto (yama1066) . . . 3
2.5 tanykazy . . . 3
2.6 -80486- . . . 3
2.7 Noritada Kobayashi (noritada) . . . 3
2.8 jsynth21 . . . 3
2.9 TANIGUCHI Takaki (takaki_t) . . . 3
2.10 dancerj . . . 3
2.11 YAMASHITA Junji (ysjj) 3
2.12 koedoyoshida . . . 3
2.13 Kouhei Maeda (mkouhei) . 3

3 Debian での Rust 4
3.1 Debian ユーザとして Rust プログラムを書いてみる . . . 4
3.2 Docker で使うには . . . 5
3.3 Debian Developer としての Rust パッケージ . . . 5

4 メモ 7



1　最近の Debian 関連のミーティング報告

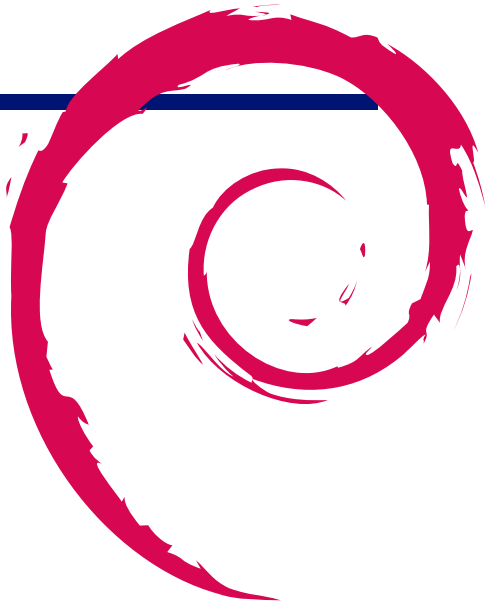
杉本　典充

1.1　2020 年 9 月度 東京エリア・関西合同 Debian 勉強会

2020 年 9 月 19 日 (土) に東京エリア Debian 勉強会と関西 Debian 勉強会の合同でオンラインによる Debian 勉強会を開催しました。参加者は 10 名でした。

セミナーは「DebConf 20 のイベント共有会」と題して参加者でイベントの情報共有を行いました。共有内容は以下の URL に記録しました。

`https://gobby.debian.org/export/l10n/ja/20200919-tokyodebian`



2 事前課題

杉本　典充

今回の事前課題は以下です。

1. Rust でプログラムを書いたことがありますか（Hello world でも可）。
2. 次のプログラム言語でどのように debian パッケージが作られているか興味があるものを教えてください。

2.1 dictoss

1. 書いたことはありません
2. golang, nodejs, python

2.2 yy_y_ja_jp

1. 書いたことはありません
2. golang, nodejs

2.3 NOKUBI Takatsugu (knok)

1. 書いたことはありません
2. emacs, golang, nodejs, その他

2.4 Hiroyuki Yamamoto (yama1066)

1. 書いたことはありません
2. rust, nodejs, ruby, python, その他

2.5 tanykazy

1. 書いたことはありません
2. vim

2.6 -80486_

1. 書いたことはありません
2. (回答なし)

2.7 Noritada Kobayashi (noritada)

1. 書いたことがあります
2. emacs, vim, golang, rust, nodejs, perl, ruby, python

2.8 jsynth21

1. 書いたことはありません
2. perl, python, その他

2.9 TANIGUCHI Takaki (takaki_t)

1. 書いたことがあります
2. emacs, rust, nodejs, perl, ruby, python

2.10 dancerj

1. 書いたことがあります
2. emacs, nodejs, python

2.11 YAMASHITA Junji (ysjj)

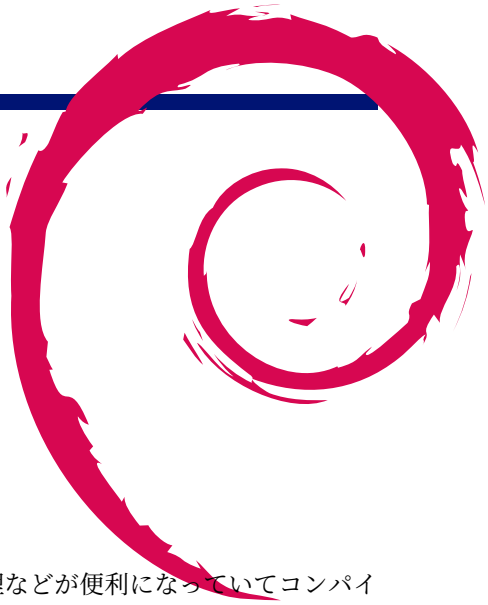
1. 書いたことがあります
2. rust

2.12 koedoyoshida

1. 書いたことはありません
2. python

2.13 Kouhei Maeda (mkouhei)

1. 書いたことがあります
2. emacs, vim, rust



3 Debian での Rust

上川純一

Rust[2] というプログラミング言語をご存知ですか？私は C++ よりメモリ管理などが便利になっていてコンパイルさえ通ると大体メモリの所有権についての問題が解決しているということで勉強してみることにしました。

執筆時点では Version 1.47.0 が公開されており、比較的安定しているようです。

良い解説書 [1] がオンラインにあり、日本語にも活発に翻訳 [3] されているようです。

しかしどうも Debian には Firefox のビルドに必要だからパッケージされているだけのような気がしなくもない。

3.1 Debian ユーザとして Rust プログラムを書いてみる

まず入門書を読み始めるとおもむろに Rust のサイトからバイナリパッケージをインストールするためにシェルスクリプトをウェブからダウンロードしてきて実行するように書いてあります。

```
$ curl --proto '=https' --tlsv1.2 https://sh.rustup.rs -sSf | sh
```

しかしインストールの副作用は何があるのかとかを確認したいし、アンインストールするときに何をしたら良いのかなど考えるのが面倒ですよね。そこで Debian ユーザとしてはパッケージを利用したい。Debian のパッケージを利用するにはどうしたらよいかを解説します。

```
$ sudo apt install cargo rustc
```

まず Buster では最初にリリースされたときは 1.33[*1]だったのですが２０２０年９月に比較的新しい 1.41.1[*2]にアップデートされました。firefox-esr のセキュリティーアップデートの依存関係だったようです。The Rust Book[1] が 1.41.0 を前提としているのでまぁこれでいいんじゃないでしょうか。

3.1.1 エディタの設定

Emacs を使っているのなら elpa-rust-mode を入れておきましょう。

```
$ sudo apt install elpa-rust-mode  # for local emacs.
```

Vim は `https://github.com/rust-lang/rust.vim#installation` にたくさん書いてあるのですがちょっとよくわかってません。

3.1.2 シンプルなプログラムを書く

チュートリアルに従ってプログラムを書いてみます。

[*1] 2019 年 2 月 28 日にリリース

[*2] 2020 年 2 月 27 日にリリースされているようです `https://blog.rust-lang.org/2020/02/27/Rust-1.41.1.html`

```
fn main() {
   println!("Hello, world!");
}
```

コンパイルして実行してみます。

```
$ rustc helloworld.rs
$ ./helloworld
Hello, world!
```

3.1.3 Cargo を使う

Rustc を直接つかうよりは Cargo というビルド管理ツールを使うことが多いようです。テンプレートの生成からビルドやテストの実行までやってくれます。Crate とよばれる Rust のパッケージシステムが利用できるようになります。

Cargo 関連のファイルはホームディレクトリ以下の`~/.cargo/` に作成されます。

cargo new でプロジェクトのテンプレートが作成され、Hello world とだけ出力する src/main.rs が生成されます。cargo build でビルド。cargo run で実行、cargo test でテストが実行されます。

```
$ cargo new 名前
$ cd 名前
$ cargo build
$ cargo run
Hello, world!
$ cargo clean
```

Cargo.toml というファイルに依存関係を記述することができます。この Crate パッケージインストールは Debian の管理下ではなくユーザのホームディレクトリ以下にインストールされます。

3.2 Docker で使うには

Docker イメージで使うということもあるでしょう。公式で提供されている Docker イメージがあるのでそれを使うのが手っ取り早いと思われます。Debian のパッケージを使いたい場合はパッケージをインストールすればよいです。

```
FROM debian

RUN apt-get clean && apt-get update && apt-get install -yq \
      make \
      rustc \
      && apt-get clean
```

しかしながら Dockerhub にはすでに公式の Docker イメージが提供されており、rust パッケージは Debian Buster に Rustup で Rust の最新版をインストールしたものになっています [7]。パッケージとして提供されていないのは気になりますが Docker イメージの中なのでアンインストールの方法などを考える必要がなく、影響範囲が限定されており、便利に使えそうです。

```
FROM rust
```

3.3 Debian Developer としての Rust パッケージ

Rust パッケージをメンテンスしているチーム [4] がいるので Wiki ページなどに情報がまとまっています。Crate のパッケージングについての資料 [5] に解説があります。

パッケージのためのツールとして debcargo というものを使うようです。

```
$ apt update && apt install debcargo
```

おそらく一番の困難は、Rust の Crate は複数のバージョンが共存できるように作られているのですが、Debian 側のポリシーとしては単一のバージョンのライブラリをできるだけ使いたいです。そうすると古いバージョンを要求される場合はそれをできるだけ新しいバージョンで置き換えることになるでしょう。大量にパッチを書くことになりそ

うです。

debcargo の動作原理ですが、Cargo の利用する cargo.toml ファイル [6] に依存関係の情報があるのでそれを利用して Debian パッケージを生成するようです。

参考文献

[1] https://doc.rust-lang.org/book/

[2] https://www.rust-lang.org/

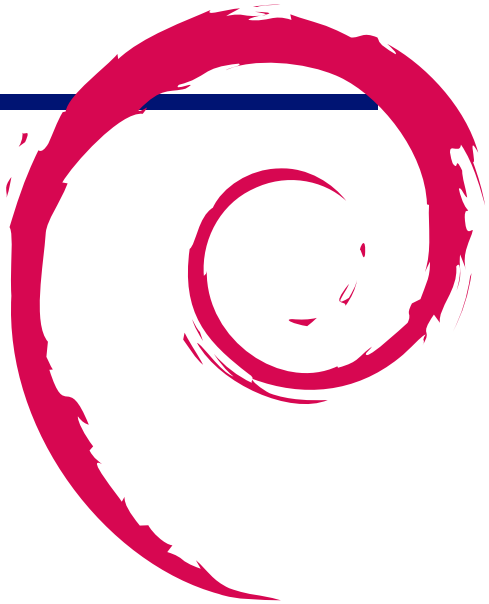
[3] “Rust の日本語ドキュメント/Japanese Docs for Rust” https://doc.rust-jp.rs/

[4] https://wiki.debian.org/Teams/RustPackaging

[5] “Packaging a crate for Debian” https://salsa.debian.org/rust-team/debcargo-conf/blob/master/README.rst

[6] “The Manifest Format” https://doc.rust-lang.org/cargo/reference/manifest.html

[7] “Rust – Docker Official Images” https://hub.docker.com/_/rust



4 メモ

Debian 勉強会資料

2020 年 10 月 17 日　　初版第 1 刷発行

東京エリア Debian 勉強会 （編集・印刷・発行）