

ラズパイ的技術系同人誌 執筆のすゝめ

[著] あいおん

c101 新刊

2022 年 12 月 31 日 ver 1.0

■免責

本書は情報の提供のみを目的としています。

本書の内容を実行・適用・運用したことで何が起こりようとも、それは実行・適用・運用した人自身の責任であり、著者や関係者はいかなる責任も負いません。

■商標

本書に登場するシステム名や製品名は、関係各社の商標または登録商標です。

また本書では、™、®、©などのマークは省略しています。

はじめに

前書き

♣ 書くに至った経緯

こんにちは。あいおんです。お久しぶりの人はお久しぶりです。はじめましての人ははじめまして (n 年ぶり n 回目)。何作目かわからず数えてみたところ 8 作目だそうです。末広がりですね。縁起がいい気がします。

さて今回久しぶりに筆を取ったわけですが、どんな感じで書こうかなと迷ってました。というのも、コロナの影響でコミケ等のイベントに出品するのも久しぶりでして、もともとどうやって書いてたかとか思い出すところからです。そもそもどうしてまた書こうかと思ったかという、c100 の影響です。友人と久しぶりに参加者として参加したんですが、そこで創作意欲を掻き立てられて・・・そうして気づけば c101 の参加用紙をもらっていました。そんなこんなで応募したところまたもや運よくスペースをいただけたので、久しぶりに書くかという気合を入れて今書き始めています。

♣ 書くにあたって

今までは RaspberryPi をベースに私自身インフラエンジニアであるので、サーバ用途としてのお話を書いてました。また、一通りサーバ構築についてシリーズを書きましたら、今度は趣味でフロントエンド周りが楽しく思え、これでも執筆はしました。そして、今回。さていよいよネタも尽きてきたなぁという頃でした。最初に本を出したのが 2016 年で、もう 6 年もたつわけですから Windows も 7 から 10/11 にステップアップしまして PC も変えました。となると、乗っかっているソフトウェアも総入れ替えになるわけですし、そうした時にもともと執筆で使っていた Word がいない環境となっております。ギリギリ前回執筆した時は、旧マシンを引っ張り出してきてやったわけなんですけど、ちょっともうその体力もなく、せっかく技術書を書くのにいつまでも Word2007 では良くないなぁという思いもあったので、この機会に新執筆環境をこさえてみようと思ったのが、この本を書くきっかけとなっております。そういうわけで、今回からレイアウト等が異なっていると思うのですが、そのあたりはご容赦いただければと思います。

実はこのテーマにする前は「SE としての心構え」を書くつもりでいました（ので、サークル紹介にはそう書いてある箇所もあるかもです・・・）。が、よくよく考えたら、権威があるわけでも何でもないどこの馬の骨ともわからない奴がいう言葉なんて誰が読みたいんだと思ったのでやめました。ちょうど執筆環境を新調するし、今までやってたラズパイも絡めて技術要素入れれば本として成立するかな、と思ってできたのがこの本であります。そういうわけでそんな感じで今回の本の内容もご理解いただければと思います。

♣ 本書について

目的

ということでこの本の目的は「ラズパイで技術系同人誌の執筆環境を整備すること」です。技術本なんて別に書く気ないよ、という方でもラズパイを使った環境整備ということでちょっと立ち寄っていただけると嬉しいです。

対象

本書では以下のような人に特におすすめです。当てはまらない人にもおすすめです。

- ラズパイを持っていて、使い道に困っている人
 - いろんな仕組みを使って執筆環境作ってみますので、スキルアップにつながります。
- いろんな人の技術本を読んで面白いなと思っている人
 - 自分も生み出す側のその一人になってみませんか
- 自分のスキルのアウトプットをしたい
 - 本を作るのは非常に良い機会です。

できるようになること

この本を読み終わるころには以下のスキルを習得していることでしょう。

- 本を作るスキル
 - 原稿を作って印刷するというスキルもそうですが、文書力も身につくと思います。
- Latex/ReVIEW
 - LaTeX 自体を扱うというよりは Re:VIEW というソフトを扱う上で LaTeX の記法も勉強する必要があります。
 - 印刷できる仕組みや記法についても触れますのでこれもできるようになります。
- Gitlab
 - ラズパイにて、Git のリポジトリを Gitlab にて構築し、活用していきます。
 - Gitlab CI/CD
 - * Gitlab に Push されたときに自動的に処理を行うことができます。今回はこの機能を使ってファイル生成等を行います。
 - Gitlab Pages:
 - * Gitlab の機能で HTML のページなどを公開することができます。今回はこの機能を使って、執筆した文書をクイックに確認するようにします。
- Docker
 - コンテナ技術である Docker についても導入や利用をします。本格的な活用とまではいきませんが、最低限の利用はできるようになると思います。
- Textlint
 - 文書校正ツールである Textlint を利用して自動的にレビューをしてくれる仕組みを構築できるようになります。

前提知識

突然ですが、私は初心者向けの資料が好きです。難しいことを理解する、ということよりも、一つの物事についていろんな観点での理解をしたい人間なので。というわけで、この本もそうでありたいと思うため、前提知識は不要です。どんな人でも読んでもらえるように意識します。

一方で、内容が簡単すぎるかということそうならないようにも意識します。この本を読み終わるころにはラズパイの有効活用方法が一つ身に着けられる、というゴールを達成できるようにします。

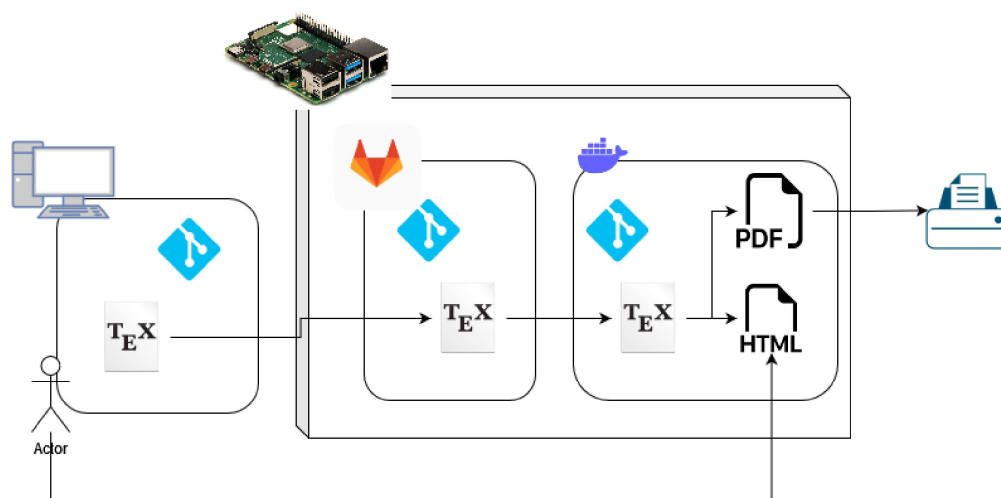
また、読んでいるときに迷子にならないようかなり丁寧にターミナルのログを記載しつつ説明しています。なので、躓いたところがあれば少し戻って一つずつ確認しながら進んでいただければおそらく同じことができるのではないかなと思っています。

構築する環境

今回この本で構築していく環境は図 1 を参照してください。イメージとしては以下の流れで執筆を行います。

1. 自分のいつも使ってる環境 (ローカル環境) で LaTeX に似た機能の Re:VIEW 記法で執筆する。
2. ローカル環境で執筆したら Git でバージョン管理を行う。
3. ある程度区切りがつかいたらコミットし、ラズパイの Gitlab リポジトリに Push する。
4. Gitlab は Gitlab CI/CD を設定しておき、Push されたら自動的に Docker を起動し、PDF と HTML を生成する。
5. HTML に関しては Gitlab Pages の機能を利用し、ローカル環境から内容を確認できるようにしておく。
6. Textlint が CI/CD の中で実行され、自動的に校正が実施される。
7. 執筆が完了したら PDF ファイルを回収して印刷する。

このようにして自動執筆環境を構築することを目標とします。



▲ 図 1: 全体構成図

前提

♣ 前提知識

ラズパイのシリーズではもはやおなじみになりましたが、今回も書いておきます。今回利用するラズパイについてですが、基本的にはサーバとしての利用となり、GUI(グラフィカル=画面)で使うことはなく、CUI(キャラクター=文字)での操作になります。なので、ラズパイはあらかじめOSのインストールやSSHの設定を完了しておいてください。すみませんが、ここから書き始めるとこれまた一冊かかってしまうので、この部分が不安な方は私の1冊目を読んでいただきたく思います。ここでは、コンソールの軽い操作説明等をしておきます。以下、Teratermでログインして、操作した時のログを例に解説します。ログインの仕方もまとめておきますので、参考にしてください。TeratermとはWindowsでSSHをするときのデファクトスタンダードともなっているようなソフトです。フリーですのですぐに使うことが出来ます。Macでしたら、OS標準のTerminalで出来ます。

▼ 表 1: ログイン情報

項目	値
IP アドレス	rp.local
UserName	pi
Password	raspberry

▼ リスト 1: ターミナル基本画面説明

```
pi@rp:~ $                               ← Username@Servername:CurrentDirectory
pi@rp:~ $ pwd                             ←現在のディレクトリを表示
/home/pi
pi@rp:~ $
pi@rp:~ $ cd /home                        ←移動 (/homeディレクトリに移動)
pi@rp:/home $
pi@rp:/home $ ls  ←内容を表示(ディレクトリの中身を一覧表示)
pi
pi@rp:/home $
pi@rp:/home $
```

主要なコマンドである表 2 は押さえておきましょう pi ユーザーは、あらかじめ「sudo」コマンドの権限が付与されています。よって、「sudo su -」を実行することによって root ユーザーになれるので覚えておきましょう。今回利用する OS は Raspberry Pi OS です。(Linux raspberrypi 5.4.83-v7l+) 特に理由がなければ同じものを使っていただけると良いかと思います。

また、ビット数については 32bit とします。(まだまだ 64bit 対応のものが多くないので、こちらで行きます。どうせメモリも 4GB なので。ただし、SD カードは最低 32GB ぐらいを用意していただければと思います。おそらく足りると思いますが 64GB が推奨です。(Docker 等を利用する

ので比較的使います。) そんなに値段も変わらないと思うので、64GB 以上だと何かと便利かと思っていますので、参考にしていただければと思います。

▼ 表 2: 主要なコマンド

コマンド	意味
pwd	PrintWorkDirecotry であり、カレントディレクトリを表示する。
cd	ChangeDirectory であり、カレントディレクトリを移動する。
ls	LiSt であり、カレントディレクトリの中身を一覧表示する。
cat	conCATenate であり、ファイルの内容を表示する。(連結する)
cp	CoPy であり、ファイルをコピーする。
rm	ReMove であり、ファイルを削除する。
mkdir	MaKeDirectory であり、ディレクトリを作成する。
file	ファイルのタイプを確認する
vi	VI エディターでファイルを編集する。(ここは emacs でも可です。)
sudo	SuperUserDO であり、管理者として実行する
apt-get	ラズパイにソフトウェアをインストールするときに利用します。
docker	Docker 用のコマンド
rake	Ruby のビルドコマンド
gem	Ruby で利用するパッケージをインストールするときに利用します。
gitlab-ctl	Gitlab の制御用コマンド
reboot	再起動コマンド
git	git でバージョン管理する場合に利用するコマンド
openssl	証明書などを生成できるコマンド
curl	Web からファイルをダウンロードするコマンド

また今回からターミナルやファイル系の記載は以下のようにしています。

▼ リスト 2: ターミナル系の説明

```
pi@rp:~/reviewDocker $ docker images
REPOSITORY      TAG                IMAGE ID           CREATED            SIZE
review          haranoaji         724eefcc3cf6      About an hour ago  2.9GB
review          latest            724eefcc3cf6      About an hour ago  2.9GB
debian          bullseye-slim     77c993218a5e      11 days ago       74.3MB
hello-world     latest            46331d942d63      9 months ago      9.14kB
pi@rp:~/reviewDocker $ ← haranoajiというタグが付いた
```

入力をしてもらうようなコマンドについては薄い下線が引いてあります。また、特に確認してほしい出力については下線や太線になっています。

▼ リスト 3: ファイル系の説明

```
RUN apt-get update && \
    apt-get install -y --no-install-recommends \
    nodejs && \
```

```
apt-get clean && \  
rm -rf /var/lib/apt/lists/* && \  
npm install -g yarn textlint && \ ←追記  
npm install -g textlint-plugin-review textlint-filter-rule-comments \  
               textlint-filter-rule-whitelist \  
               textlint-rule-preset-ja-technical-writing ←追記
```

ファイル系についてもターミナル系と同様ですが、薄い文字で書かれているものはコメントであり記載する必要はありませんので注意してください。また、本の幅の特性上改行を複数個所に入れていることがあります。基本的にはそのような箇所には「\」が記載されていますので、必要に応じて端折ってもらっても構いません（\をして改行する代わりに、そのまま\を入力せず、改行もせず入力しても基本的には同じ結果になります。）

♣ 正誤表について

内容は確認していますが、万が一誤りがあった場合は以下のリポジトリにて訂正をしていきますので、あれ？ と思うところがあったら確認をお願いします。

また、改善点や要望についてもあれば以下リポジトリの Issues を上げていただければ真摯に対応いたします。

<https://github.com/aion0721/c101-errata>

目次

はじめに	i
前書き	i
構築する環境	iii
前提	iv
第 1 章 技術本を執筆する意味	1
1.1 なぜ技術本を作るのか	1
♣ 私の書くきっかけ	1
♣ 今までのテーマ	2
♣ 教訓	5
1.2 本を作る流れ	7
♣ 参加申し込み	7
♣ 本を作る	7
♣ イベント出店	9
1.3 この章のまとめ	11
[コラム] キーボード自作してみました	12
第 2 章 RaspberryPi で執筆環境構築	15
2.1 前提環境	15
♣ OS セットアップ	15
♣ 初期設定	15
2.2 Re:VIEW	20
♣ インストール	20
♣ チュートリアル	22
[コラム] rake pdf って何?	25
2.3 Docker	26
♣ なぜ Docker か	26
♣ インストール	26
♣ Docker コンテナイメージビルド	28
[コラム] トラブル発生時の対応	29
♣ Docker コンテナでファイルをビルドする	31
♣ 別環境の Docker コンテナイメージをビルド	32
[コラム] あなたの知らないフォントの世界	35
2.4 Gitlab	37
♣ Git と Gitlab-CE	37
♣ インストール	38

	♣ リポジトリを立てる	40
	♣ HTTPS 化	42
	[コラム] 暗号化通信	44
	♣ Gitlab-Runner	45
	♣ Gitlab-Pages	48
2.5	textlint	50
	♣ Textlint 対応 Docker イメージを作成	50
	♣ パイプラインに組み込む	51
2.6	この章のまとめ	52
	[コラム] Re:VIEW Starter	52

あとがき		53
------	--	----

ラズパイ的技術系同人誌執筆のすゝめ

2022 年 12 月 31 日 ver 1.0 (c101 新刊)

著 者	あいおん
発行者	あいおん
連絡先	philosophy.aion@gmail.com https://aion.jp.net @rudolf_aion (https://twitter.com/rudolf_aion)
イラストレーター	らい
印刷所	さくら製本

© 2022 あいおん

(powered by Re:VIEW Starter)