.Deb

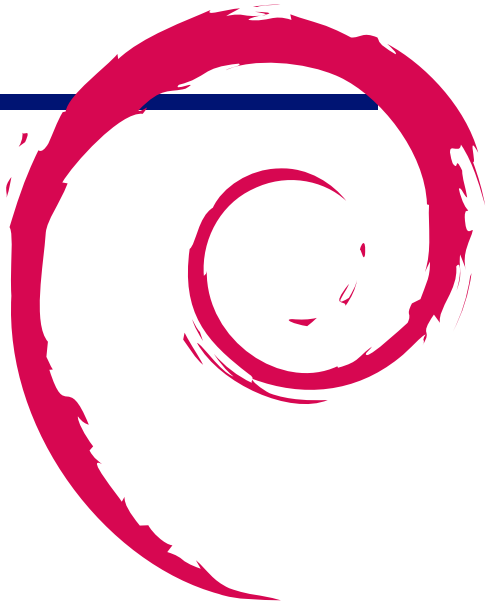
銀河系唯一のDebian専門誌

2021 年 1 月 16 日

nodejs 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

杉本 典充

1.1 2020 年 12 月度 東京エリア・関西合同 Debian 勉強会

2020 年 12 月 20 日 (日) 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 17 名でした。

セミナーは上川さんによる「Debian 勉強会の資料の作成方法について」、やまねさんによる「実演：レガシーな Debian パッケージをモダンにしてみる」の発表、および参加者全員参加による BoF「2020 年の振り返りと来年の目標」を行いました。

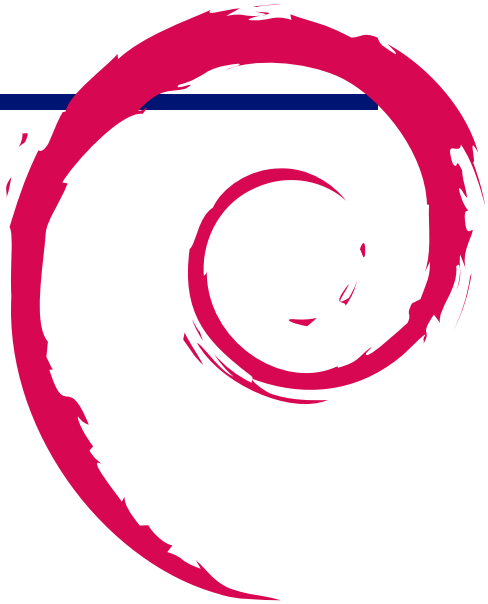
「Debian 勉強会の資料の作成方法について」の発表では 2020 年に取り組んだ Debian 勉強会の資料のビルドシステムの変更を踏まえたビルド環境の構築方法、原稿資料の作成方法、git の操作の説明をしました。ビルドシステムの変更では、ファイルの文字コードの UTF8 への変更、ビルドの高速化、一度のビルド対象を小さくするためのサブディレクトリへのファイルの再配置、CI を使ったビルドなど多数の機能改善が行われました。

「実演：レガシーな Debian パッケージをモダンにしてみる」の発表では Debian パッケージを最新のフォーマットに対応することで得られるメリットを紹介しました。debian/control ファイルの Vcs レコードの書き方、debian/rules ファイルの簡略化な記法や hardening の効果を得る方法、ファイルのライセンスを機械可読できる形で書く方法、Salsa CI との連携を説明しました。説明の後にライブデモを行い、lintian-brush などのパッケージのメンテナンスに便利なツールを駆使したモダンなメンテナンス方法を紹介しました。

BoF「2020 年の振り返りと来年の目標」では 2020 年の Debian や世間の動向を踏まえて過去・現在・未来で何が起こったか何が起こるかについて参加者で意見を出し合いました。参加者から出た意見を次のファイルにまとめています

`https://tokyodebian-team.pages.debian.net/2020-12_tokyodebian_bof.txt`

その後、Debian の次期安定版「bullseye」のリリースに向けた進捗の確認、Debian の最近の動向について情報交換を行いました。



2 事前課題

杉本　典充

今回の事前課題は以下です。

1. node でプログラムを書いたことありますか、何を書きましたか
2. node で書かれたプログラムで Debian にあったらいいなというのはなにかありますか？

2.1 dictoss

1. 書いたことがあります
2. ・anthy で絵文字を変換できるようにしたい

2.2 uwabami

1. 書いたことはありません
2. DD

2.3 NOKUBI Takatsugu (knok)

1. 書いたことがあります
2. neologd-package ITP, コーパス、データセット整備

2.4 yosuke_san

1. 書いたことはありません
2. 検討中

2.5 Hiroyuki Yamamoto (yama1066)

1. 書いたことはありません
2. 検討中

2.6 dancerj

1. 書いたことがあります
2. 検討中

2.7 yy_y_ja_jp

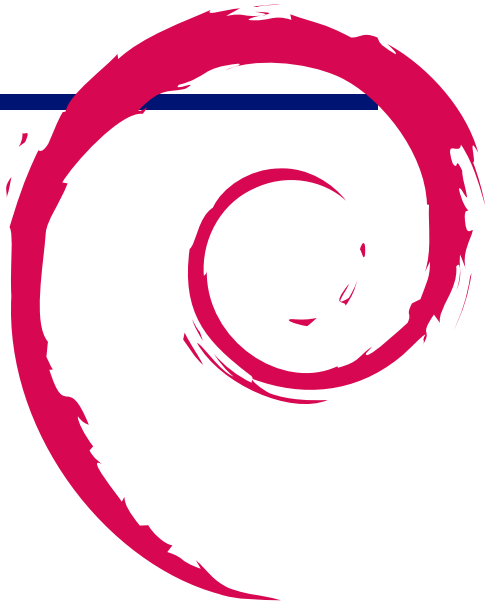
1. 書いたことはありません
2. 検討中

2.8 hashimotosyuta

1. 書いたことがあります
2. Linux の IM 周りの状況を整理したいです

2.9 ysaito

1. 書いたことがあります
2. 検討中です



3 Debian での Node

上川純一

3.1 node.js の最近

Javascript はウェブのプログラミング言語として広く使われています。node.js は v8 という Javascript エンジンで動作する Javascript 環境です [1]。v8 は Chrome の javascript エンジンとして広く使われていおり、同じ環境をブラウザ以外でも使えるというのが便利なポイントです。

Javascript は基本的にはシングルスレッドで動くのでイベントベースでプログラミングすることになり、はじめにそこをうまく扱った標準ライブラリを整備したあたりが普及の決め手だったのでしょう。たとえばほぼすべてのファイル API がコールバック形式になっています。

Javascript の言語は ECMAScript という名称で仕様化されています。その仕様は ES6 で大きく進化しました。ES6（ES2015）以降は仕様の呼称も変わり、年が入るようになりました。そして毎年仕様がリリースされています [5]。大きな変化は Promise と async/await という言語の進化により大きく書き方が変わり、それにともなって順次コアの API も書き換わっていっている過程のように見えます。

ES6 以前の Javascript だと callback を使うのでなにかするたびに function が一段かまされインデントが深くなっていきます。

```
var fs = require('fs');

fs.readFile('./cat.js', {encoding: 'utf-8'}, function(err, data) {
    if (err) throw err;
    console.log(data);
})
```

ES6 以降の Javascript だと async await が使えるようになっていて

```
const fs  = require('fs');

const main = async () => {
    const buffer = await fs.promises.readFile('./cates2015.js',
                                              {encoding: 'utf-8'});
    console.log(buffer);
}

main().catch((e)=>{
    console.error(e);
    process.exit(1);
});
```

あと javascript で必ずはまる変数スコープとか this の扱いとかを比較的意識しなくて良くなるのが良いと思います。

ここ数年は Node のバージョンは偶数バージョンが LTS の安定版となっていて、一年に一回リリースされています [2]。ES2015 から ES2020 の各機能のサポート状況については Node.js ES2015 Support [6] のページがよくまとまっているようです。

nodejs リリース	リリース日	end of life	上川にとって特筆する言語機能
v10	2018-04-24	2021-04-30	v8 6.6, fs/promises の試験導入
v12	2019-04-23	2022-04-30	v8 7.4
v14	2020-04-21	2023-04-30	v8 8.1, ES Modules の標準サポート、
v15	2020-10-20	2021-06-01	
v16	2021-04-??		

二年に一回安定版リリースをしている Debian としては一年に一回大きなアップデートをしている Node.js の最新版をおいかけるのは結構大変でしょう。

3.2 Debian ユーザとしての Node でプログラムを書いてみる

3.2.1 Debian 公式パッケージのインストール

Debian 公式のパッケージとして nodejs パッケージが提供されています。それだけでは実行できる環境が揃うだけです。Nodejs の強みは npm というパッケージリポジトリです。Debian 公式リポジトリでは npm という Debian パッケージに npm パッケージ管理システムのクライアント npm コマンドが入っています。[*1]

```
$ apt-get install -y npm nodejs
```

しかし、いくら Debian のパッケージが多いといってもすべてのパッケージを提供するのは現実的ではないのと他の nodejs を利用している一般開発者は npm 管理のパッケージを利用しているためそことの相互運用のことも考えると悩ましいですが Debian パッケージより node.js 公式パッケージを使うことになりそうです。

npm install して依存関係をインストールすると Debian パッケージを使うわけではなく、`http://npmjs.com` からインストールすることになります。

Debian パッケージとして依存関係をインストールなら該当するパッケージを探すことになります。インストールされたライブラリは /usr/lib/nodejs 以下にインストールされています。

手元の Socket.io テストアプリの例を見てみましょう。

```
{
  "name": "wssrtc",
  "description": "Node.js socket.io app running on Cloud Run for webrtc",
  "version": "v0.0.1",
  "private": true,
  "dependencies": {
    "cli": "",
    "ejs": "",
    "express": "",
    "socket.io": ""
  },
}
```

Debian パッケージは node-express などだと思われるんですが、cli, socket.io にそれぞれ該当するパッケージが見つかりませんでした。昔はあったような気がするのに。

npm install を使うと `npmjs.com` からダウンロードしてきて `./node_modules` 以下にインストールします。しかしなんか Node.js のバージョン 10 がサポートされませんと警告が出たり微妙な雰囲気です。

```
$ npm install
npm WARN npm npm does not support Node.js v10.21.0
npm WARN npm You should probably upgrade to a newer version of node as we
npm WARN npm can't make any promises that npm will work with this version.
npm WARN npm Supported releases of Node.js are the latest release of 4, 6, 7, 8, 9.
npm WARN npm You can find the latest version at https://nodejs.org/
npm WARN deprecated debug@4.1.1: Debug versions >=3.2.0 <3.2.7 || >=4 <4.3.1 have a low-severity ReDos regression when used in a Node.js envi
npm WARN ws@7.4.2 requires a peer of bufferutil@^4.0.1 but none is installed. You must install peer dependencies yourself.
npm WARN ws@7.4.2 requires a peer of utf-8-validate@^5.0.2 but none is installed. You must install peer dependencies yourself.

added 96 packages from 107 contributors in 3.568s
```

[*1] yarn というのもあるよと教えてもらいましたが今後の課題

3.2.2 Node.js 公式パッケージのインストール

Node.js から提供されているパッケージのインストール手段としては公式に配布されているバイナリの最新の LTS 版を利用するのがよいでしょう [3]。ダウンロード方法について書いてあるページに従うとバイナリインストールから始まるのですがそこからパッケージ版へのリンクがあり、Debian 等は別のページにあり、Debian パッケージのインストールに到達するには 3 ページ遷移する必要があります。Debian パッケージのインストール手順に従うと、サイトからダウンロードしたシェルスクリプトを root 権限で実行することになります。

```
# curl -sL https://deb.nodesource.com/setup_14.x | bash -
# apt-get install -y nodejs
```

スクリプトでやっていることは基本的には次のような一連のコマンドでした。

```
$ curl -sSL https://deb.nodesource.com/gpgkey/nodesource.gpg.key | sudo apt-key add -  # 鍵の追加
$ VERSION=node_14.x
$ DISTRO="$(lsb_release -s -c)"
$ echo "deb https://deb.nodesource.com/$VERSION $DISTRO main" | sudo tee /etc/apt/sources.list.d/nodesource.list  # パッケージ情報の追加
$ echo "deb-src https://deb.nodesource.com/$VERSION $DISTRO main" | sudo tee -a /etc/apt/sources.list.d/nodesource.list  # ソースパッケージ情報
$ sudo apt update && sudo apt install nodejs  # インストール
```

ここまでの作業を行うと npm コマンドや nodejs コマンドが利用できるようになっています。Debian 公式パッケージでは npm コマンドがわかれていますが、nodejs 提供のパッケージでは npm コマンドも nodejs パッケージに含まれています。

Debian では以前 node コマンドがすでに存在していたのでポリシーに基づいて node という名前がつかえず、nodejs という名前を利用しています、stretch までは node-legacy パッケージをインストールしてシンボリックリンクを用意していました[*2]。現在は node コマンドが利用できるようになっており、nodejs もシンボリックリンクとして提供されているようです。

node.js の API のドキュメント [4] を参照しながらプログラムを書くことになるでしょう。

3.3 Docker での Debian ユーザとしての Node の利用

Docker を利用する場合は、Node.js がインストールされたインスタンスが Dockerhub にあるのでそれをベースに使うのがよいのではないでしょうか。`https://hub.docker.com/_/node/` にあり、`FROM node` と何も指定しないと `node:current` が使われて、それは debian stretch ベースのイメージのようです。buster が使いたい場合は `FROM node:buster` を指定するのがよいでしょう。

手元の Dockerfile では次のように書いていました。

```
FROM node:14

COPY . .
RUN npm install
```

3.4 Debian Developer としての Node.js パッケージ

Debian パッケージを Nodejs の `package.json` から生成すること自体は簡単です。npm2deb をつかえばよいでしょう。おそらく困難は Debian ポリシーに従うと Debian パッケージとして追加するには Debian パッケージだけで依存関係を解決する必要があります。そのため依存関係を全部追加し、ライセンスなどのチェックを行うことにあると思われます。あとは npm 管理じゃない環境にスナップショットを維持することになるので面倒が増えます。

[*2] `https://lists.debian.org/debian-devel-announce/2012/07/msg00002.html` に名前衝突についての CTTE Resolution があります

3.4.1 Node.js バージョン

Debian 公式のパッケージ「nodejs」のバージョンを２０２０年１２月２５日時点で確認したところ、最新の LTS 版を experimental に入れるという運用のようです。保守的ですが２年間安定版として利用されることを見越すとこれが現実的でしょう。最新の Node.js の機能に依存しているソフトウェアは扱いが難しそうです。

Debian release	nodejs バージョン
stretch	4.8.2
stretch-backports	8.11.1
buster	10.21.0
bullseye	12.19.0
sid	12.19.0
experimental	14.13.0

3.4.2 Node.js パッケージポリシー

Node.js パッケージのポリシーが書いてある Wiki ページがありました [7]。そこにはこんなことが書いてあります：

- バイナリーとソースパッケージは node-foo という名前にする
- MUST: nodejs に Build-Depends を指定すること
- /usr/lib/nodejs/ か /usr/lib/nodejs/foo/にインストールすること。場所は複数のファイルがあるのか単一ファイルなのかに依存する。
- package.json を /usr/lib/nodejs/foo/package.json にいれておくこと
- ウェブブラウザから利用可能であれば libjs-foo バイナリパッケージを作成する。 Javascript/Policy を参照。
- require('foo') が動くように確認する自動テストを含めること。
- c++ アドオンは nodejs-abi-*process.versions.modules* に依存すること。

`node-`ではじまるバイナリパッケージは 1400 くらいあるようです。ただ npm にあがっているパッケージの数は 140 万くらいのようで毎日平均 1000 パッケージが追加されているようです。

3.4.3 Debian Javascript policy

歴史的には Debian にある既存の Javascript のコードの多くはブラウザで利用できるようになっているもので、それについては Debian Javascript Policy で統一しようとしているようです [8]。

javascript-common パッケージを Recommend してパッケージ名は `libjs-` で始めるようにするというものですが Buster では 223 パッケージあるようです。

3.4.4 npm2deb を使ってみる

ためしに npm2deb を使ってみました。公開されている npm パッケージをとってきてテンプレートを作成してくれるというものでした。しかし実際に使おうとするとものによりますが依存関係を全部確認して Debian に入れるのが大変そうです。

```
$ npm2deb create simple-peer

This is not a crystal ball, so please take a look at auto-generated files.

You may want fix first these issues:

simple-peer/node-simple-peer/debian/changelog: -- FIX_ME debian author  Sat, 16 Jan 2021 10:39:59 +0900
simple-peer/node-simple-peer/debian/control:Uploaders: FIX_ME debian author
simple-peer/node-simple-peer/debian/control:Description: FIX_ME write the Debian package description
simple-peer/node-simple-peer/debian/copyright:Copyright: 2021, FIX_ME debian author
simple-peer/node-simple-peer_itp.mail:Subject: ITP: node-simple-peer -- FIX_ME write the Debian package description
simple-peer/node-simple-peer_itp.mail:Owner: FIX_ME debian author
simple-peer/node-simple-peer_itp.mail:  Description     : FIX_ME write the Debian package description
simple-peer/node-simple-peer_itp.mail: FIX_ME: This ITP report is not ready for submission, until you are
simple-peer/node-simple-peer_itp.mail:FIX_ME: Explain why this package is suitable for adding to Debian. Is
simple-peer/node-simple-peer_itp.mail:FIX_ME: Explain how you intend to consistently maintain this package

Use uscan to get orig source files. Fix debian/watch and then run
$ uscan --download-current-version


*** Warning ***
Using fakeupstream to download npm dist tarballs, because upstream
git repo is missing tags. Its better to ask upstream to tag their releases
instead of using npm dist tarballs as dist tarballs may contain pre built files
and may not include tests.

Warnings occurred:
 [error]   get-browser-rtc: dependency node-get-browser-rtc not in debian
 [error]   queue-microtask: dependency node-queue-microtask not in debian
```

参考文献

[1] “Node.js” https://nodejs.org/

[2] “Releases—Node.js”, https://nodejs.org/ja/about/releases/

[3] NodeSource Node.js Binary Distributions https://github.com/nodesource/distributions/blob/master/README.md#debinstall

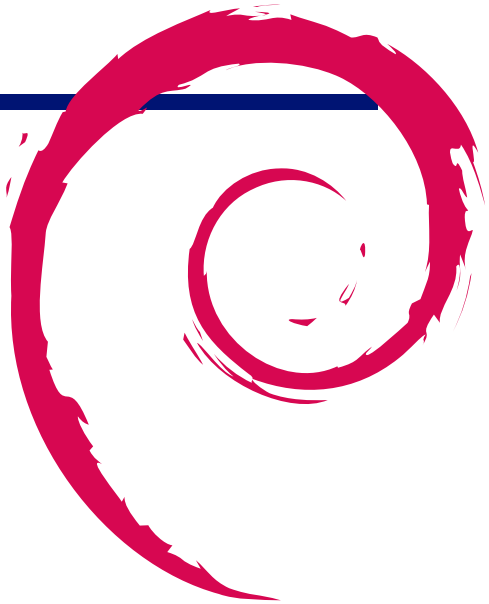
[4] “Node.js v15.4.0 Documentation” https://nodejs.org/api/

[5] “ECMAScript 2015 (ES6) とそれ以降のバージョン” https://nodejs.org/ja/docs/es6/

[6] “Node.js ES2015 Support” https://node.green/

[7] “Node.js modules policy” https://wiki.debian.org/Javascript/Nodejs/Manual

[8] “Javascript Policy” https://wiki.debian.org/Javascript/Policy



4 メモ

Debian 勉強会資料

2021 年 1 月 16 日　　初版第 1 刷発行

東京エリア Debian 勉強会 （編集・印刷・発行）