

Rmenu Raspberry PI 3B+

Rmenu 環境構築 Raspberry PI3B+

Rmenu development term

2019/07/06

目次

第1章	はじめに.....	4
第2章	構築作業、SD イメージ作成.....	5
1.	Raspbian Buster Lite をダウンロード.....	5
2.	SD メモリカードフォーマッターを取得しインストール.....	6
3.	SD カードをフォーマットします.....	8
4.	img ファイルの書き込みプログラムのインストール.....	9
5.	img ファイルの書き込み.....	9
6.	ssh ファイルの書き込み.....	10
7.	SD カードを取り付け、電源を入れる.....	11
第3章	最初のログイン.....	12
1.	ルータが付与した IP アドレスへログイン.....	12
第4章	Linux raspberrypi 初期設定.....	13
1.	ログイン.....	13
2.	パスワードの変更.....	14
3.	その他の変更.....	17
第5章	Linux 初期設定.....	18
1.	OS バージョン、ディスク容量を確認.....	18
2.	システムの更新.....	19
1.	vim を追加.....	20
1.	SWAP ファイルを 1024KB に設定.....	21
2.	一般ユーザー rmenu_user を作成する、.....	22
第6章	apt-get パッケージ管理.....	24
1.	readline-devel flex bison zlib-devel を追加.....	24
2.	日本語処理に必要な nkf コマンドをインストールする.....	27
3.	tcsh をインストールする.....	27
4.	Rmenu フォルダを RaspberryPI の rmenu_user ディレクトリに置く.....	28
第7章	Apache2 の起動.....	30
1.	次の URL をブラウザで開く http://ww.xx.yy.zz/.....	30
2.	apache2 のインストール.....	30
3.	http:// ww.xx.yy.zz / を開く.....	31
4.	アクセス権に www-data グループを追加.....	32
第8章	PostgreSQL インストール.....	33
1.	postgresql インストール.....	33
2.	postgres ユーザーにパスワードを設定.....	34
3.	TCP 接続の設定.....	34
4.	db をリストアする.....	35
第9章	RVM および Ruby のインストール.....	37
1.	RVM のインストール、(マルチユーザ) root でインストール.....	37

2.	GPG signature を取得する.....	37
3.	GPG signature を取得後、再実行.....	38
4.	pi ユーザで動作確認.....	38
5.	rvm の読込と rvm を最新バージョンに更新.....	39
6.	OS を再起動.....	39
7.	rvm を sudo を付けて (マルチユーザーとして) インストールする.....	40
8.	インストールに当たっての必要要件を確認する。.....	42
9.	続いて該当バージョン (2.6.3) をインストール.....	42
10.	最後にパスを通して、また通常使用するバージョンを指定する.....	43
11.	システム起動時の ruby デフォルトを 2.6.3 とする.....	43
12.	install されている ruby の確認.....	44
13.	関連モジュールのインストール手順.....	45
第10章	Passenger のインストール.....	50
1.	passenger のインストール手順.....	50
2.	設定の確認.....	53
3.	passenger を起動.....	53
4.	SD カード保護のため swap メモリを 0 にする.....	53
第11章	サーバ電源投入時・Rmenu 起動スクリプトの自動化.....	55
1.	起動スクリプト.....	55
2.	起動設定.....	55
3.	コンフィグ画面起動.....	55
4.	「Boot Options」を選択します。.....	55
5.	「B2 Wait for Network at boot」を選択します。.....	55
6.	<はい>を選択します。.....	55
7.	<了解>を選択します。.....	55
8.	リブートしてください。.....	55
第12章	Rmenu Application のインストールと起動.....	56
1.	rmenu_user フォルダの所有グループ・パーミッションを変更.....	56
第13章	Setubi Application 起動.....	59
第14章	注意事項.....	61
1.	Raspberry PI の性能と互換性.....	61

第1章 はじめに

Rmenu は Web アプリケーションを構築するためのフレームワークです。
業務アプリケーションを効率的に作成することを目的としています。
一般的な Linux サーバ、VPS サーバーや AWS EC2、オンプレミスの CentOS で動作させることができます。

また、Raspberry PI 3B 等のボード PC でもクワッドコアを搭載した基板で Raspbian OS でも動作します。

Rmenu は IOT の端末部のフレームワークとしても便利に利用することができます。

今回の勉強会の資料として MicroSD カードに書き込むだけで、すぐに動作確認できるように、Rmenu デモシステムの実行イメージファイルを用意しました。

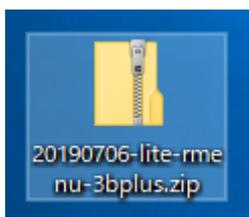
元にした RaspbianOS は
Raspbian Buster Lite
Version: June 2019
Release date: 2019-06-20

この手順書は Raspberry PI MODEL 3B+ に RaspbianOS のインストール後、Rmenu システムが動作するまでの構築手順を記述しています。

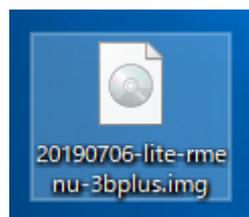
rmenu.net Web サイトより 20190706-lite-rmenu-3bplus.zip をダウンロードして頂き、解凍後
20190706-lite-rmenu-3bplus.img ファイルをマイクロ SD カードに書き込み Raspberry PI 本体に装着して頂ければ、電源起動にて動作するように用意しました。

ご自身で、構築し直したり、別な環境で Rmenu を利用したい場合の参考資料となります。

20190706-lite-rmenu-3bplus.zip ダウンローURL
<https://rmenu.net/20190706-lite-rmenu-3bplus.zip>



解凍後の img ファイル



このイメージファイルは ssh 接続が可能になっています。
ネットワークに接続後、dhcp で取得した IP をモニタ接続、ルータ等の情報により確認してください。

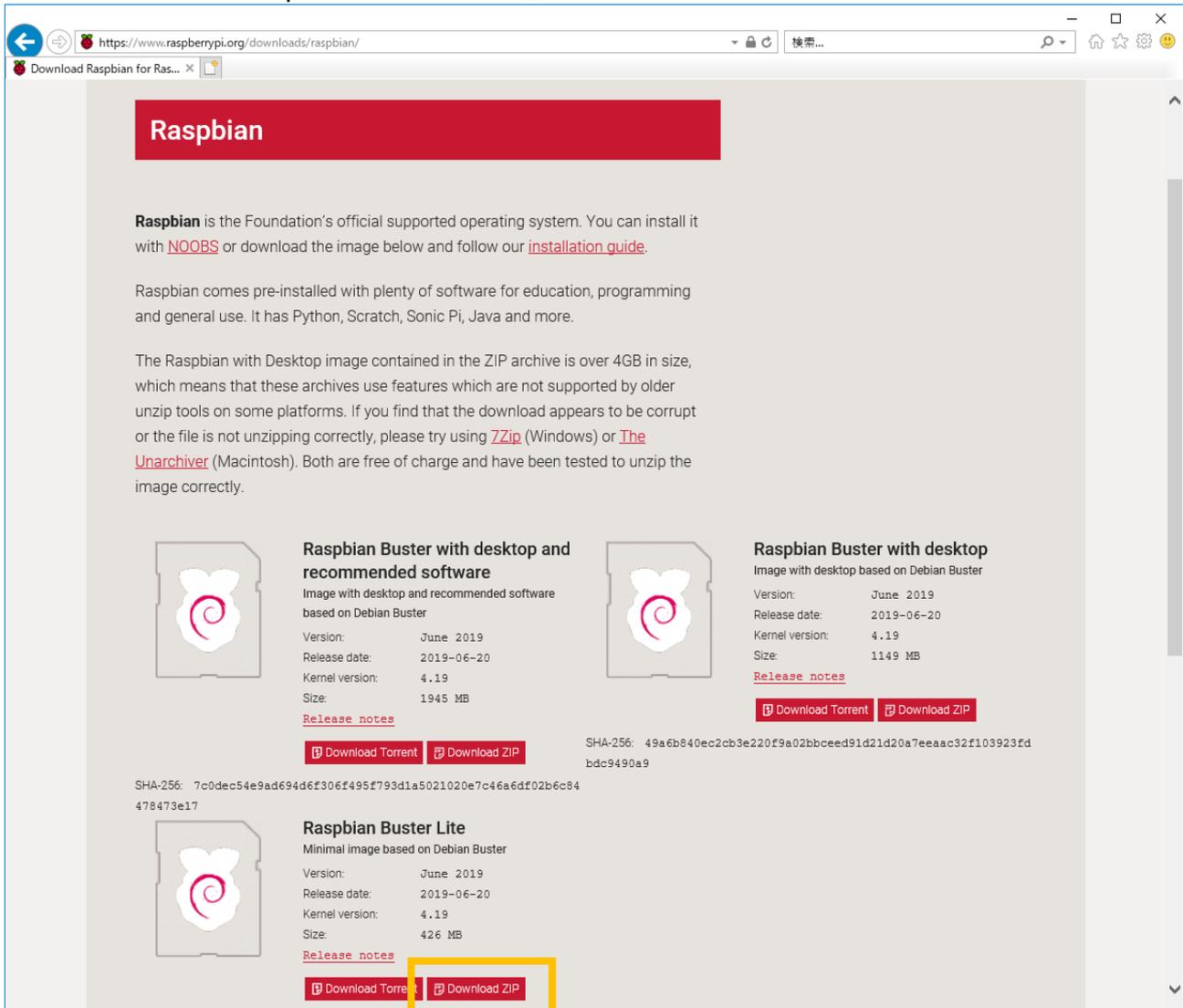
ssh 接続アカウントは次の通りです、セキュリティ確保のため、変更をしてください。

ID : pi
PW:raspberrry

Rmenu アプリケーションの動作確認は、第13章へ

第2章 構築作業、SDイメージ作成

1. Raspbian Buster Lite をダウンロード



Raspbian

Raspbian is the Foundation's official supported operating system. You can install it with [NOOBS](#) or download the image below and follow our [installation guide](#).

Raspbian comes pre-installed with plenty of software for education, programming and general use. It has Python, Scratch, Sonic Pi, Java and more.

The Raspbian with Desktop image contained in the ZIP archive is over 4GB in size, which means that these archives use features which are not supported by older unzip tools on some platforms. If you find that the download appears to be corrupt or the file is not unzipping correctly, please try using [7Zip](#) (Windows) or [The Unarchiver](#) (Macintosh). Both are free of charge and have been tested to unzip the image correctly.



Raspbian Buster with desktop and recommended software

Image with desktop and recommended software based on Debian Buster

Version: June 2019
Release date: 2019-06-20
Kernel version: 4.19
Size: 1945 MB

[Release notes](#)

[Download Torrent](#) [Download ZIP](#)

SHA-256: 7c0dec54e9ad694d6f306f495f793d1a5021020e7c46a6df02b6c84478473e17



Raspbian Buster with desktop

Image with desktop based on Debian Buster

Version: June 2019
Release date: 2019-06-20
Kernel version: 4.19
Size: 1149 MB

[Release notes](#)

[Download Torrent](#) [Download ZIP](#)

SHA-256: 49a6b840ec2cb3e220f9a02bbceed91d21d20a7eeaac32f103923fdbdc9490a9



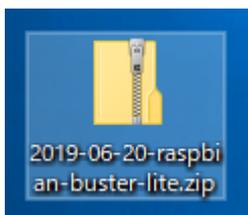
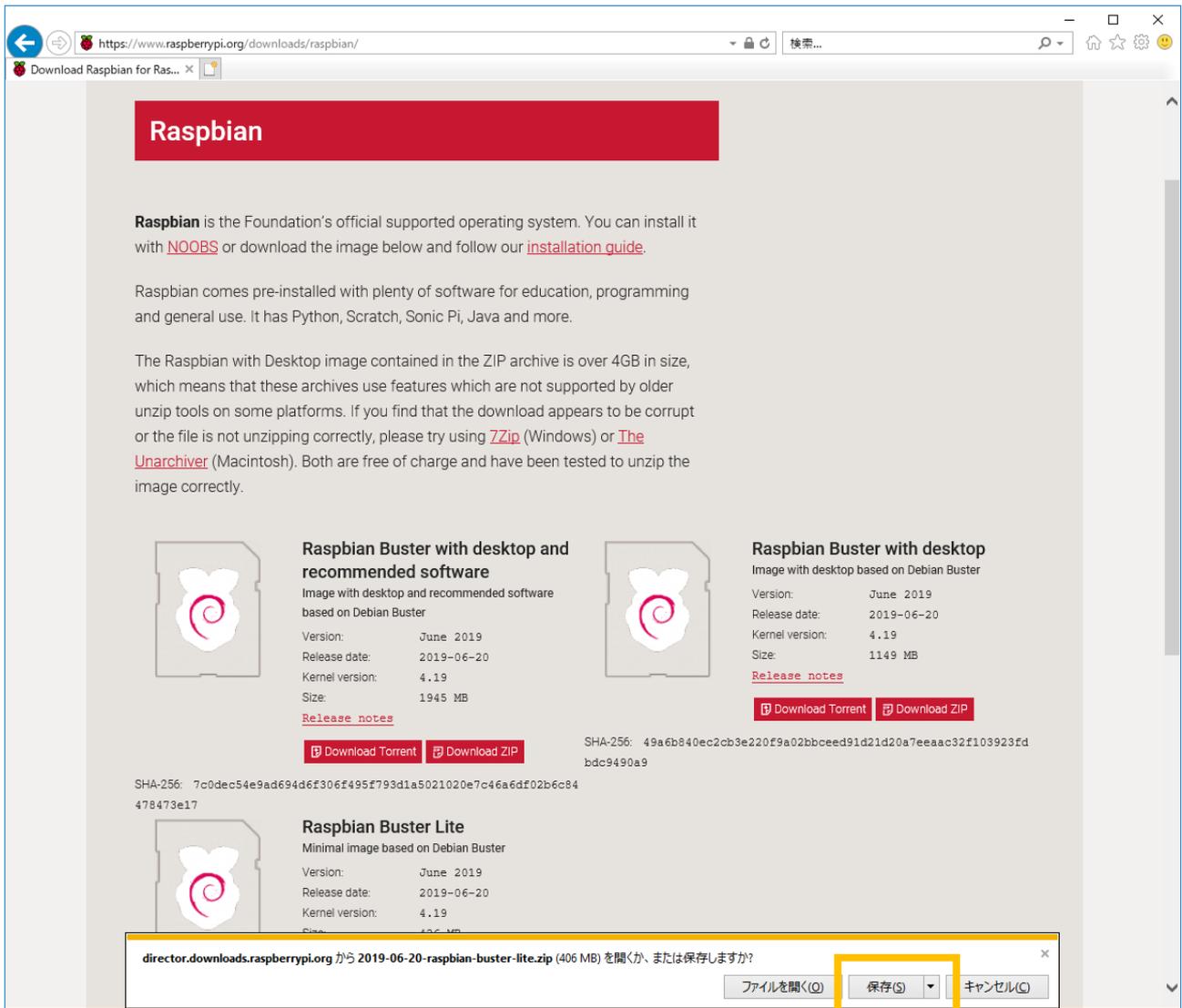
Raspbian Buster Lite

Minimal image based on Debian Buster

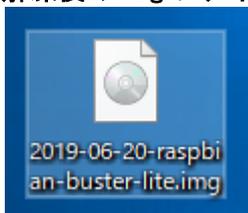
Version: June 2019
Release date: 2019-06-20
Kernel version: 4.19
Size: 426 MB

[Release notes](#)

[Download Torrent](#) [Download ZIP](#)



解凍後の img ファイル



2. SD メモリカードフォーマッターを取得しインストール

SD Card Formatter ダウンロードページ

https://www.sdcard.org/jp/downloads/formatter_4/eula_windows/index.html

SD Association

ENGLISH | 日本語 | 簡体中文 | 繁體中文

Member Login | Join the SD Association

お問い合わせ | サイトマップ

SDアソシエーションについて | ユーザー向け | 開発者向け | ダウンロード | ニュース&イベント | 会員専用サイト | 利用規約サイト

SDメモリカードフォーマッター Windows用

HOME > ダウンロード > SDメモリカードフォーマッター > SDメモリカードフォーマッター Windows用

ダウンロード

- SDアソシエーション白書
- 簡易版SD規格仕様書(English)
- SDメモリカードフォーマッター
- SDメモリカードフォーマッター Windows用**
- SDメモリカードフォーマッター Mac OS用
- よくあるご質問

<https://www.sdcard.org/jp/index.html>

SD Memory Card Formatter END USER LICENSE AGREEMENT

NOTICE:

BY DOWNLOADING, INSTALLING OR USING THE PRODUCT, THE ENTITY OR INDIVIDUAL ENTERING INTO THIS AGREEMENT AGREES TO BE BOUND BY THE FOLLOWING TERMS. IF YOU DO NOT AGREE WITH ANY OF THESE TERMS, DO NOT DOWNLOAD, INSTALL, OR USE THE PRODUCT; PROMPTLY RETURN (IF APPLICABLE) THE PRODUCT TO THE SDA OR YOUR SDA DISTRIBUTOR. IF YOU REJECT THIS AGREEMENT, YOU WILL NOT ACQUIRE ANY LICENSE TO USE THE PRODUCT.

This Agreement ("Agreement") is between the entity or individual entering into this Agreement ("You") and the SD Card Association, a California Mutual Benefit Corporation ("SDA"). "You" includes you and your employees and Affiliates. "Affiliate" is defined as an entity which controls, is controlled by, or shares common control with a party where such control may exist through ownership of securities or by contract. In addition to the restrictions imposed under this Agreement, any other

agreement, transactions and shall not apply to this agreement. This agreement constitutes the entire agreement between You and SDA and supersedes any prior or contemporaneous negotiations or agreements, whether oral, written, or displayed electronically, concerning the Product and related subject matter. No modification or waiver of any provision hereof will be effective unless made in a writing signed by both SDA and You. You may not assign or transfer this Agreement or a License to a third party without SDA's prior written consent. Should any provision of this Agreement be invalid or unenforceable, the remainder of the provisions will remain in effect. The parties have agreed that this Agreement and the documents related thereto be drawn up in the English language. Les parties exigent que la présente convention ainsi que les documents qui s'y rattachent soient rédigés en anglais.

YOU ACKNOWLEDGE AND AGREE THAT YOU HAVE READ THIS AGREEMENT AND INTEND TO BE BOUND AS IF YOU HAD SIGNED THIS AGREEMENT IN WRITING. IF YOU ARE ACTING ON BEHALF OF AN ENTITY, YOU WARRANT THAT YOU HAVE THE AUTHORITY TO ENTER INTO THIS AGREEMENT ON BEHALF OF SUCH ENTITY AND BIND SUCH ENTITY TO THE TERMS OF THIS AGREEMENT.

同意しません | **同意します**

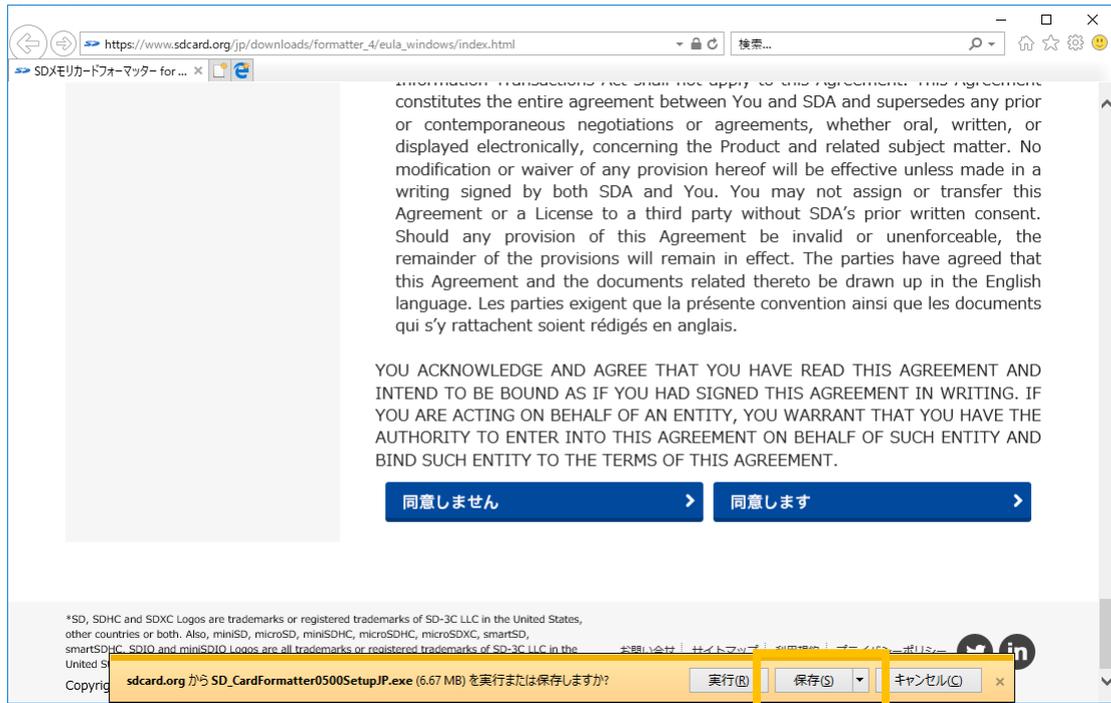
*SD, SDHC and SDXC Logos are trademarks or registered trademarks of SD-3C LLC in the United States, other countries or both. Also, miniSD, microSD, miniSDHC, microSDHC, microSDXC, smartSD, smartSDHC, SDIO and miniSDIO Logos are all trademarks or registered trademarks of SD-3C LLC in the United States, other countries or both.

お問い合わせ | サイトマップ | 利用規約 | プライバシーポリシー

Copyright (c) SD Association. All Rights Reserved.

最下部までスクロールする

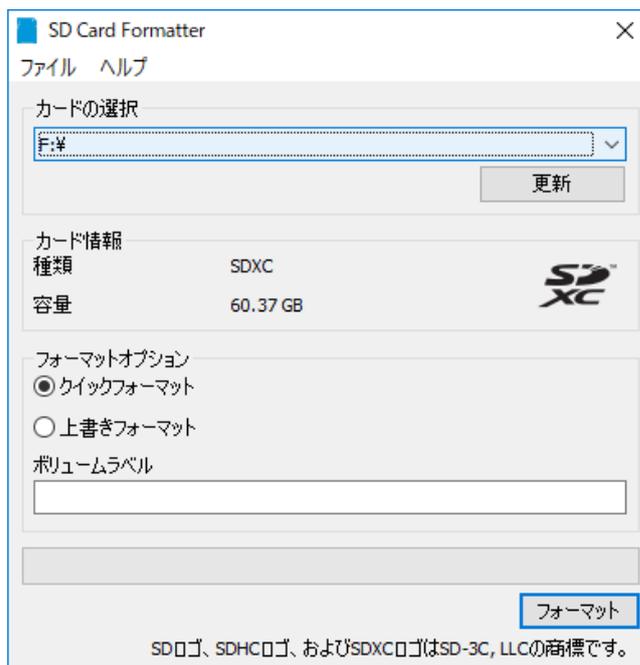
保存をクリック



ダウンロードされた、
`SD_CardFormatter0500SetupJP.exe` を実行してインストールしてください。

3. SDカードをフォーマットします

SDカードをクイックフォーマットしてください。



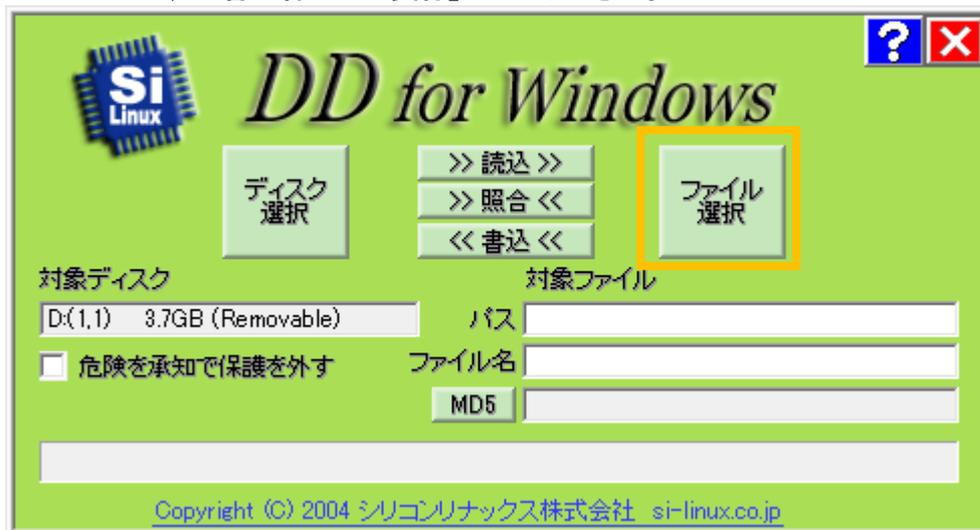
4. img ファイルの書き込みプログラムのインストール

「DD for Windows - Tech Info」を取得してインストール指定ください。
<http://www.si-linux.co.jp/techinfo/index.php?DD%20for%20Windows>

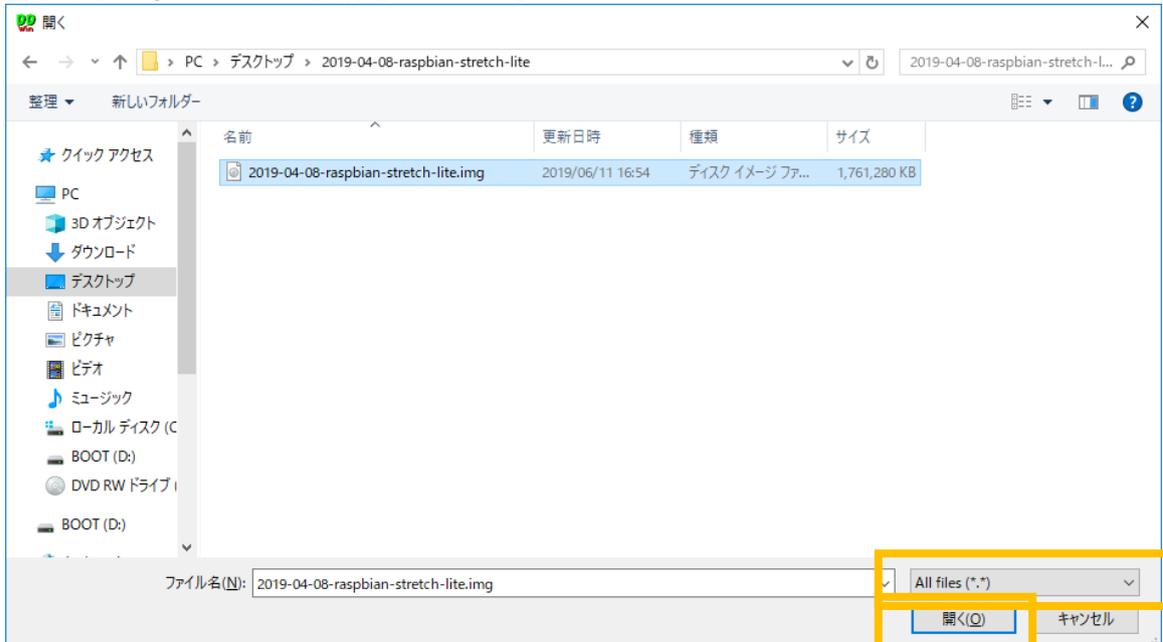


5. img ファイルの書き込み

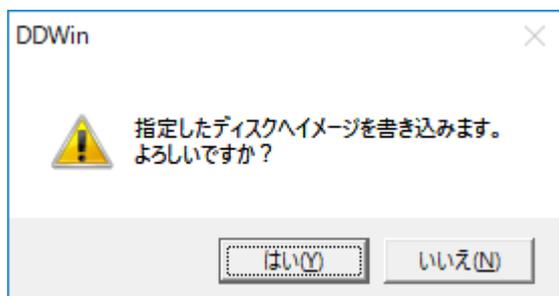
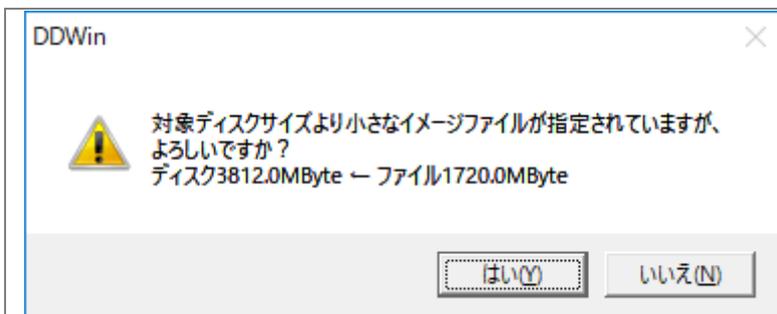
DDwin.exe は、「管理者として実行」してください。



書き込む img ファイルを指定してください。



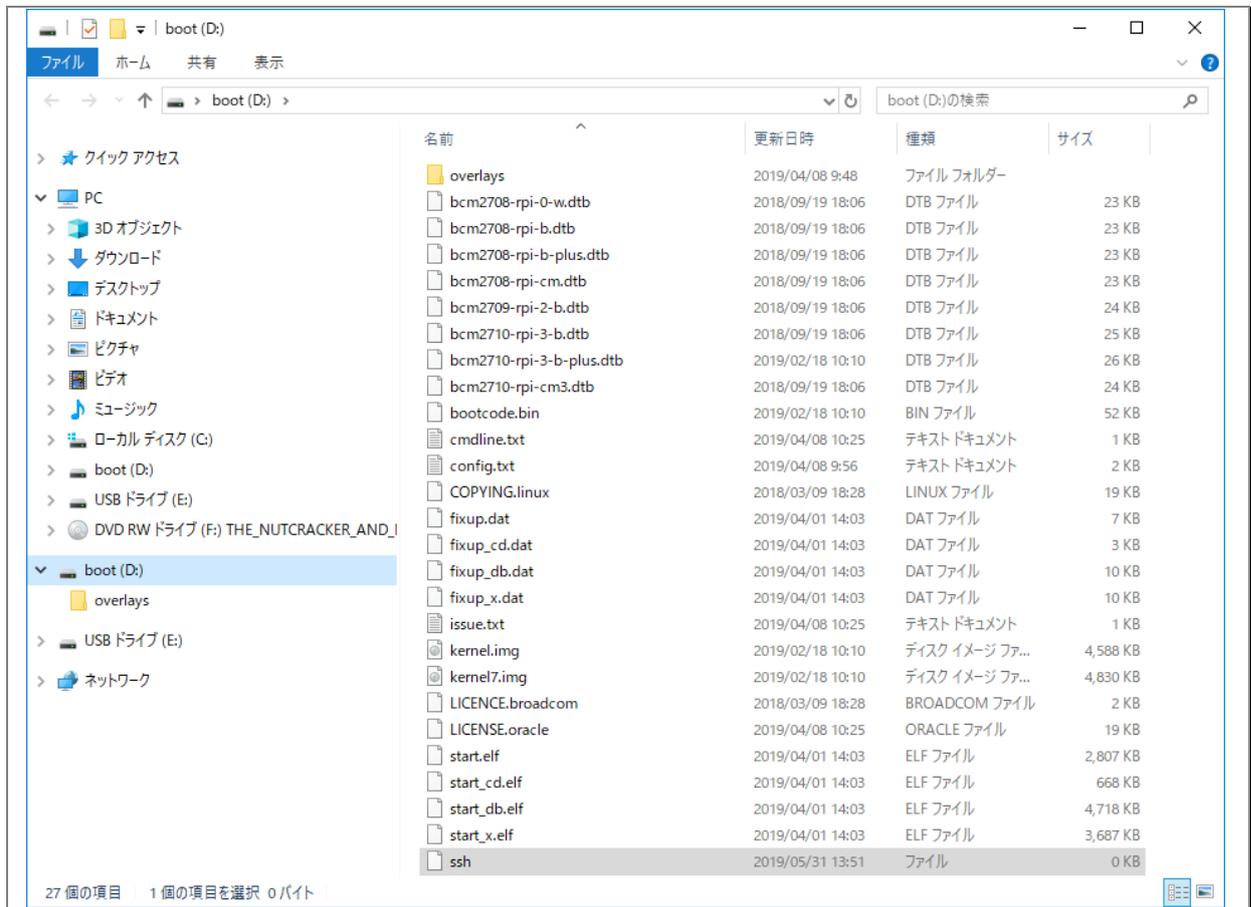
「書込」をクリック



6. ssh ファイルの書き込み

作成した SD カードのルートディレクトリにファイル名「ssh」の空のテキストファイルを新規作成してください。

拡張子は削除してください。「ssh.txt」の場合無視され ssh で接続できません。



7. SD カードを取り付け、電源を入れる

Raspberry PI に取り付けて電源を入れてください。

LAN ケーブルに接続してください。

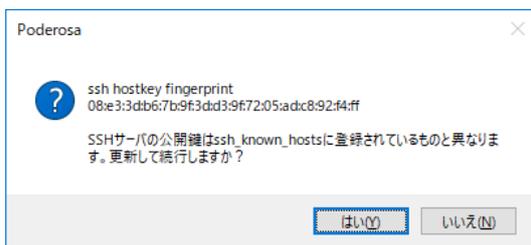
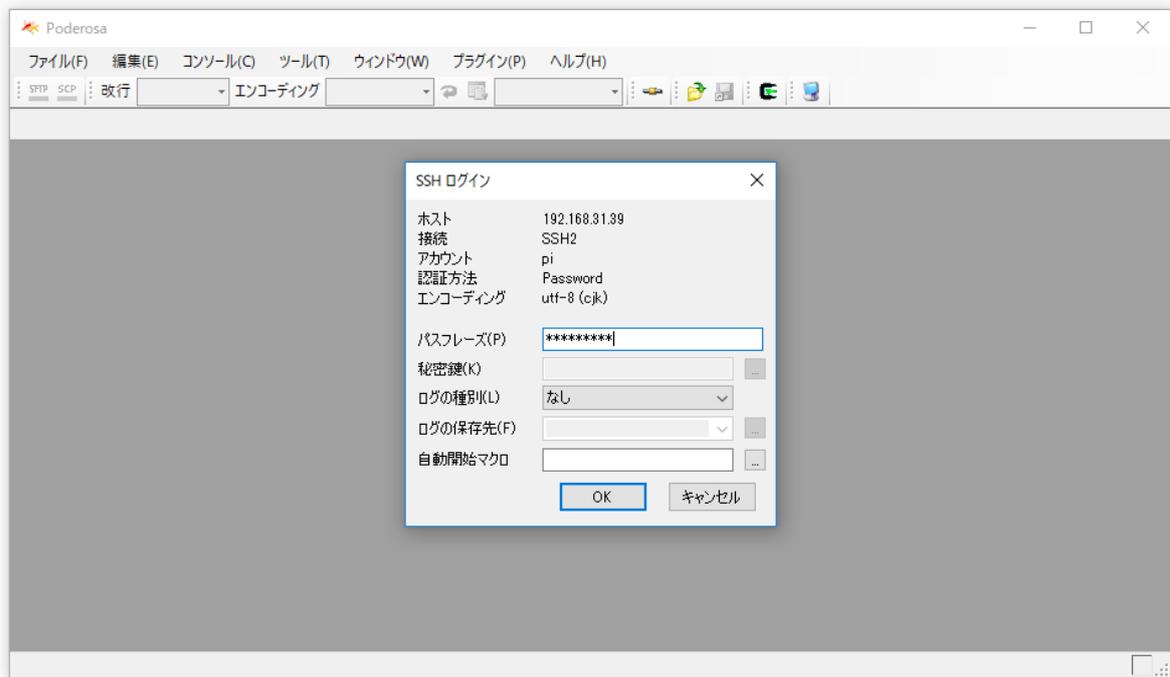
ネットワーク内に DHCP サーバーが存在する必要があります。

第3章 最初のログイン

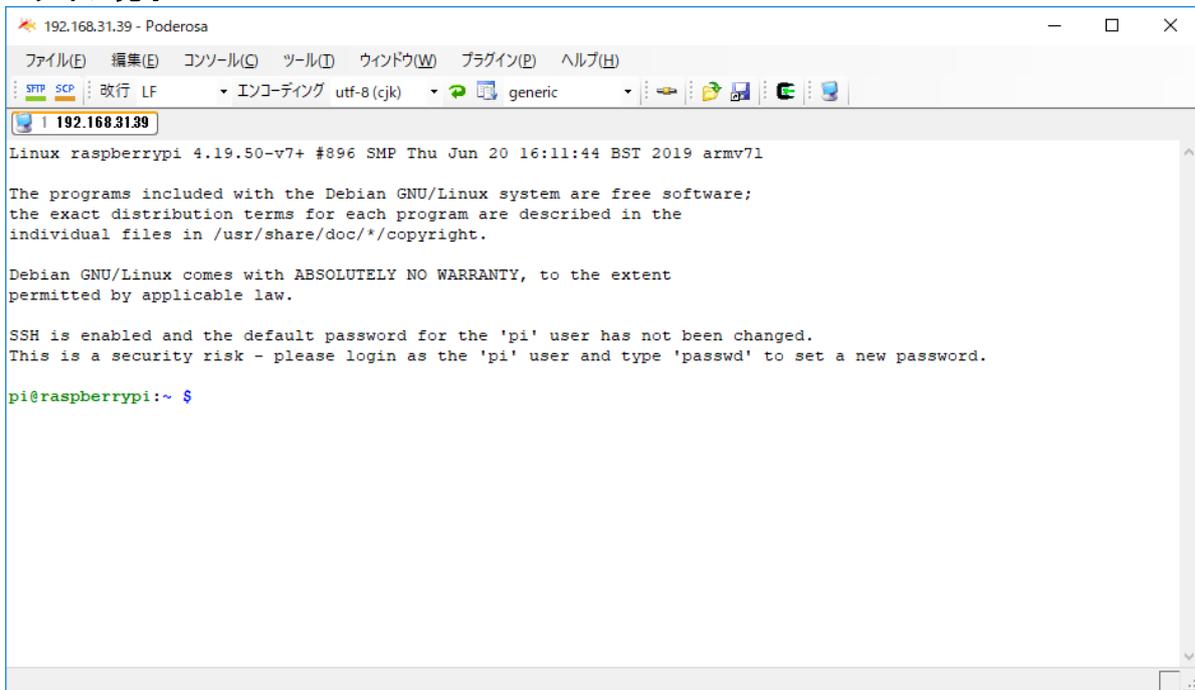
初期状態のログインパスワードは
ID:pi、PW:raspberrypi

1. ルータが付与した IP アドレスへログイン

WW.XX.YY.ZZ へログインする



ログイン完了



第4章 Linux raspberrypi 初期設定

1. ログイン

初期状態のログインパスワードは
ID:pi、PW:raspberrypi

```
Linux raspberrypi 4.19.50-v7+ #896 SMP Thu Jun 20 16:11:44 BST 2019 armv7l
```

The programs included with the Debian GNU/Linux system are free software; the exact distribution terms for each program are described in the individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent permitted by applicable law.

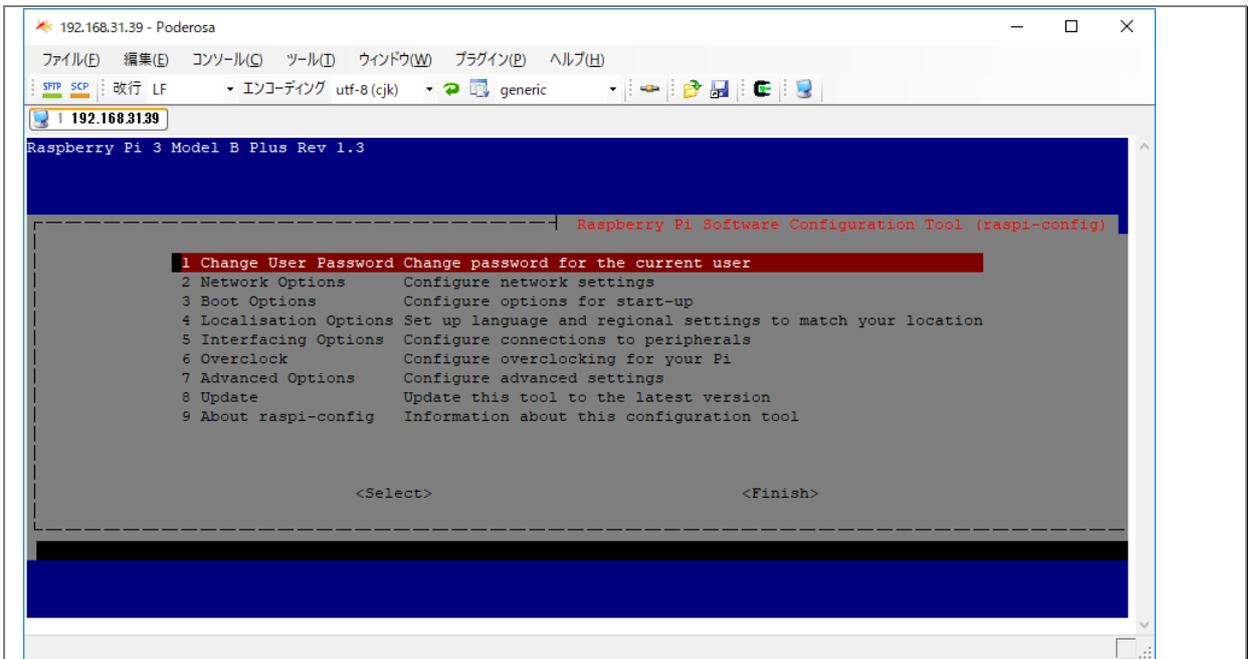
SSH is enabled and the default password for the 'pi' user has not been changed. This is a security risk - please login as the 'pi' user and type 'passwd' to set a new password.

```
pi@raspberrypi:~ $ df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/root        3.4G  1.2G  2.1G  36% /
devtmpfs         459M   0  459M   0% /dev
tmpfs            464M   0  464M   0% /dev/shm
tmpfs            464M  12M  452M   3% /run
tmpfs            5.0M   4K  5.0M   1% /run/lock
tmpfs            464M   0  464M   0% /sys/fs/cgroup
/dev/mmcblk0p1  253M   40M  213M  16% /boot
tmpfs             93M   0   93M   0% /run/user/1000
pi@raspberrypi:~ $
```

2. パスワードの変更

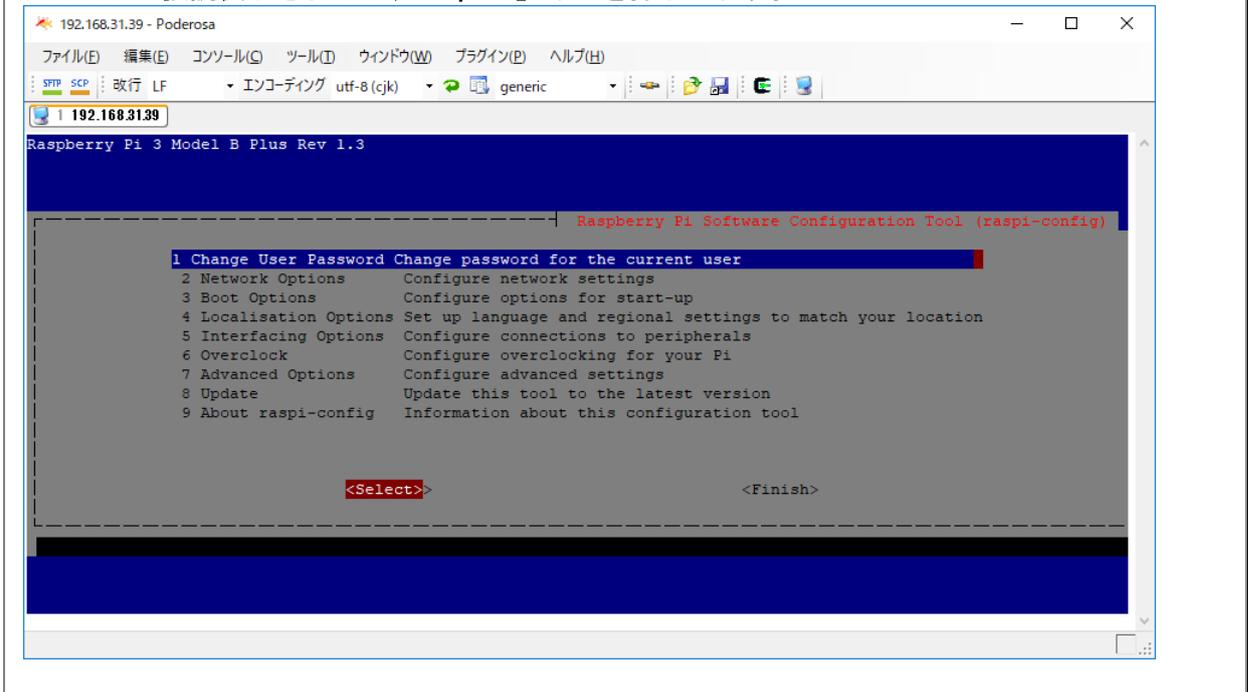
設定画面起動

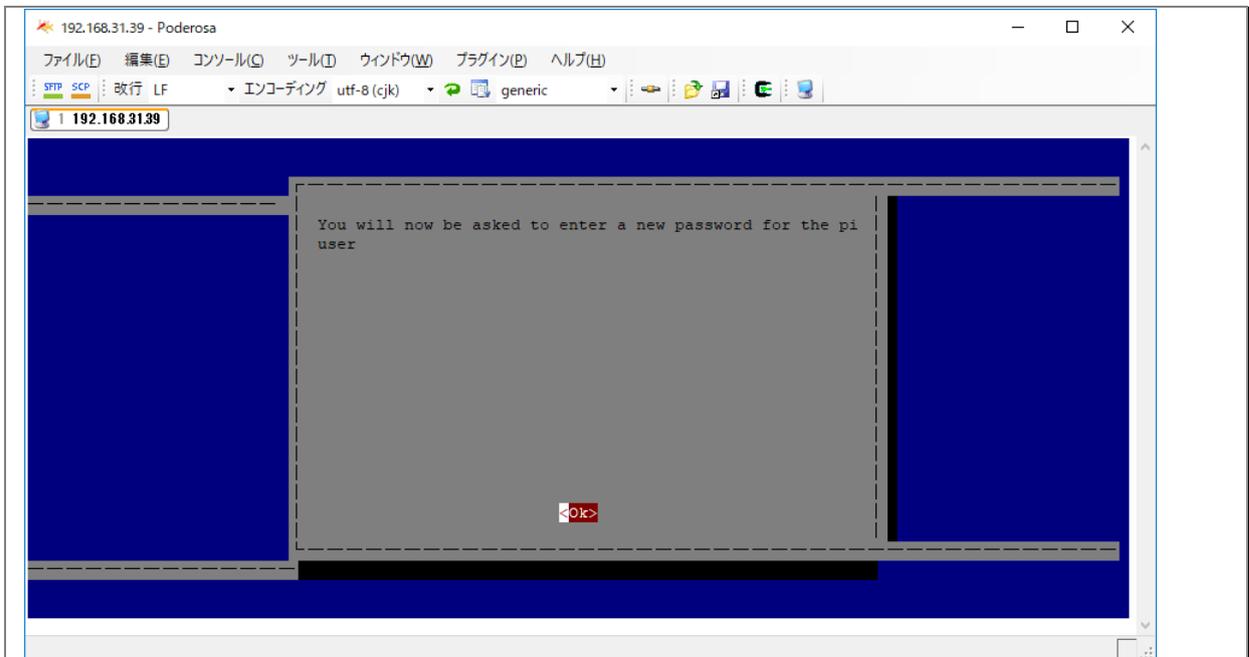
```
pi@raspberrypi:~ $ sudo raspi-config
```



「Tab」キーを押下します。

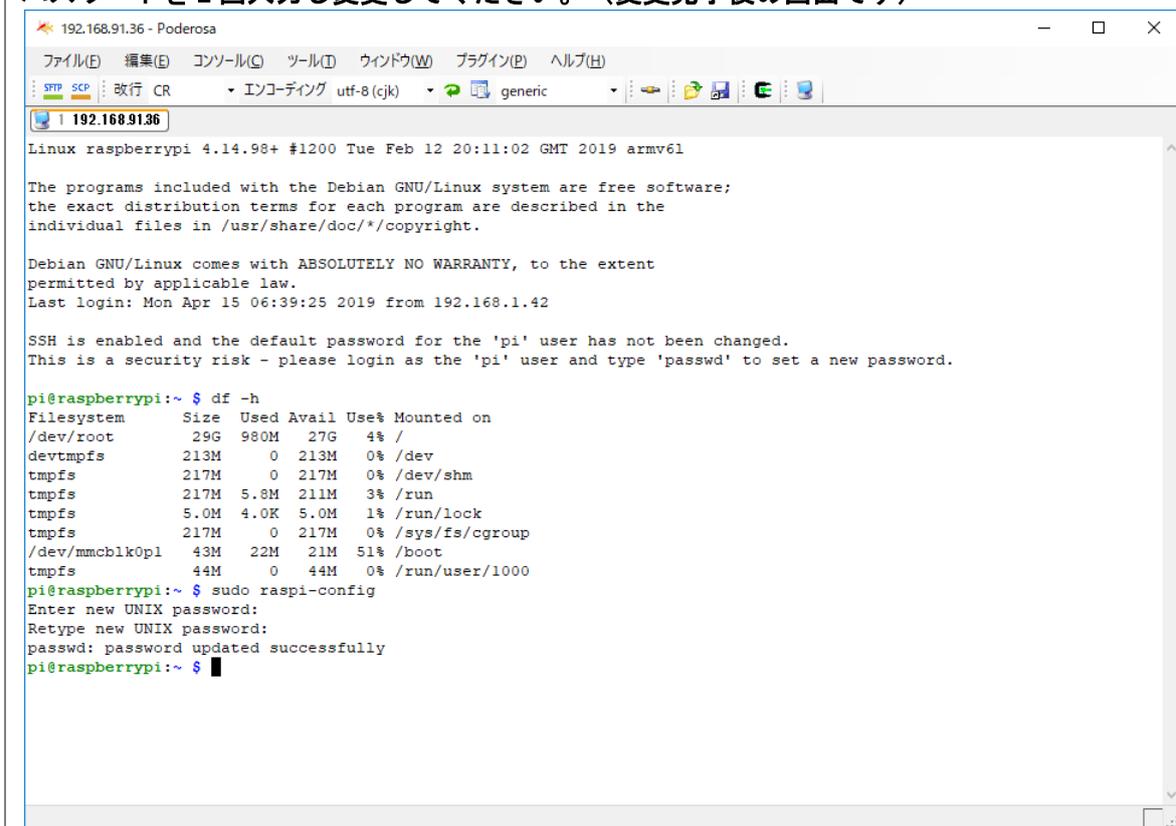
<select>が強調表示されたら、「space」キーを押下します。



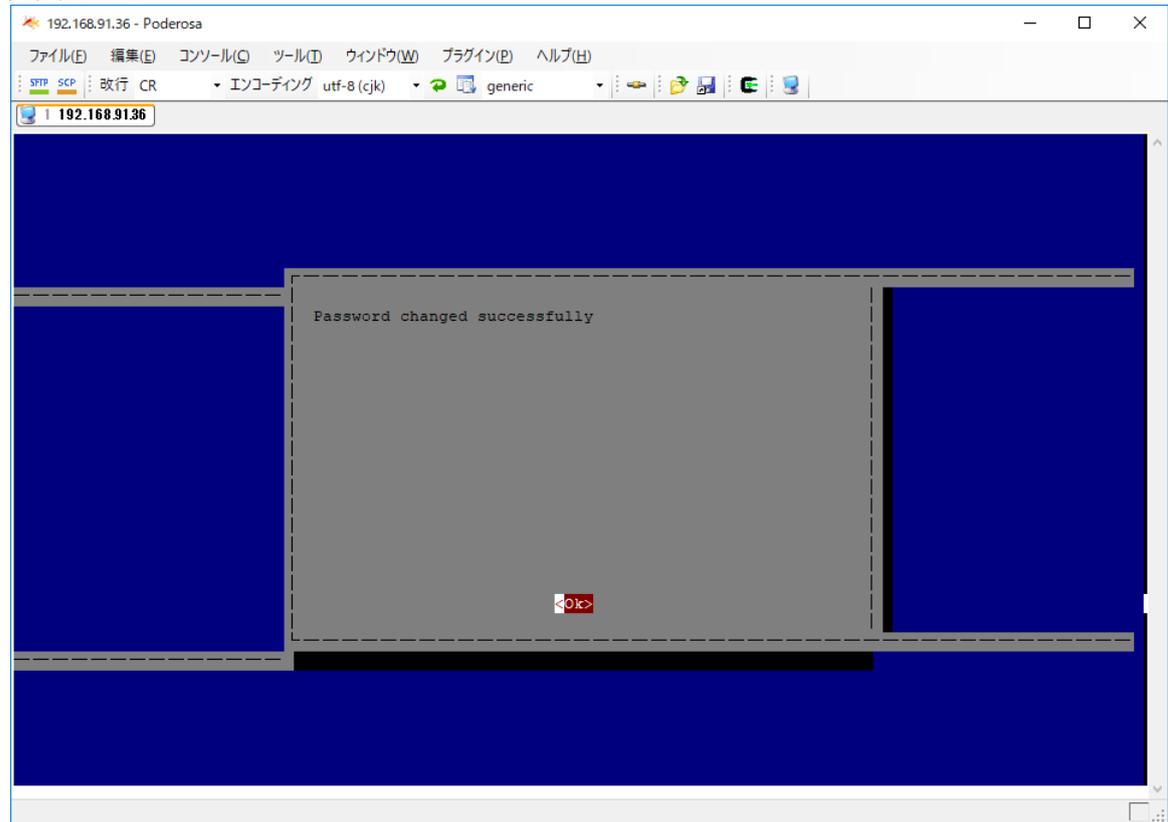


<Ok>が強調表示されたら、「space」キーを押下します。

セキュリティポリシーに基づいて任意のパスワードを設定します。
パスワードを2回入力し変更してください。（変更完了後の画面です）

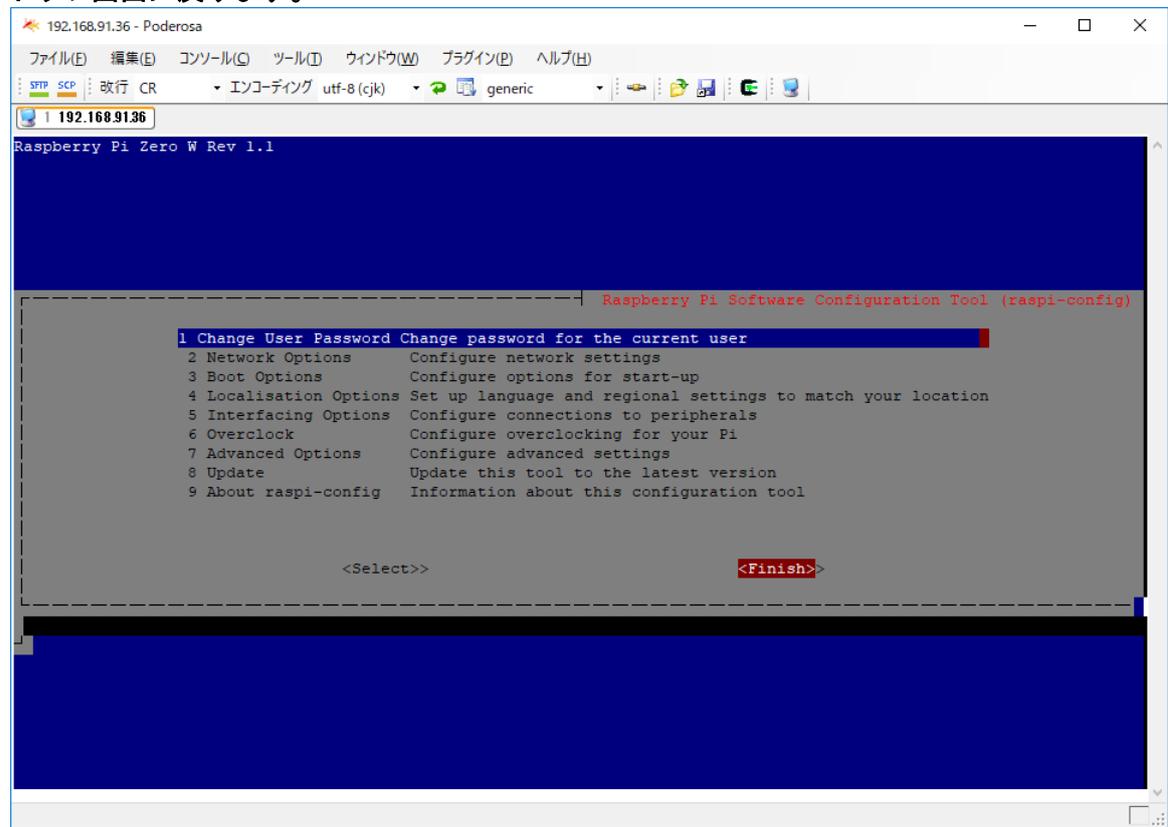


変更完了画面



<Ok>が強調表示されているので、「space」キーを押下します。

トップ画面に戻ります。



3. その他の変更

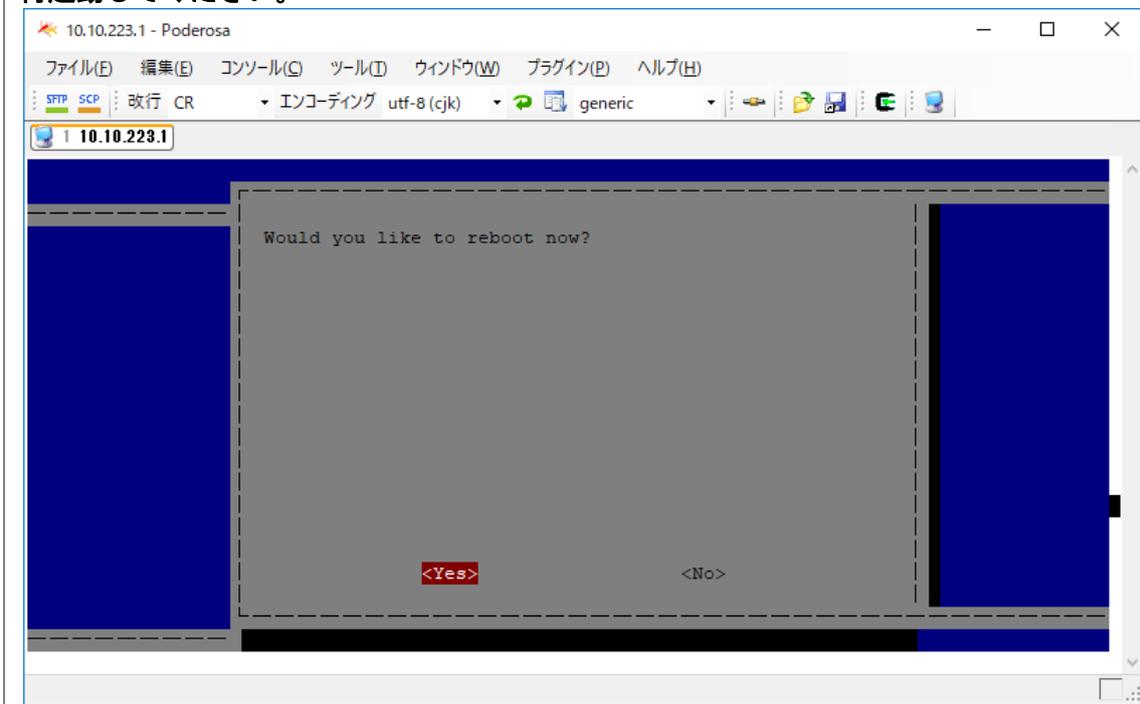
設定画面を操作し次の通り設定してください。

```
pi@raspberrypi:~ $ sudo raspi-config
```

- ・Localisation Options→Change Locale : ここは日本なので「ja_JP.UTF-8」にしている。
- ・Localisation Options→Change Locale : 「en_GB.UTF-8」を外す。
その次の画面で選べと言われるのはデフォルトのロケール、これも「ja_JP.UTF-8」でOK
- ・Localisation Options→Change Timezone : ここは日本、なので「Asia」を選んでから「Tokyo」
- ・Localisation Options→Change Wi-fi Country : Wi-Fi を使う場合は JP にしておく方が良いかと思う
- ・Advanced Options→Memory Split : GPU メモリの割り当て、ヘッドレスで運用しているのでアプリケーションが使える分を最大にすべく 16 にしている。

SD Expand しておく。

再起動してください。



第5章 Linux 初期設定

1. OS バージョン、ディスク容量を確認

初期状態のログインパスワードは
ID:pi、PW:「第4章で設定した任意のパスワード」

```
Linux raspberrypi 4.19.50-v7+ #896 SMP Thu Jun 20 16:11:44 BST 2019 armv7l
```

```
The programs included with the Debian GNU/Linux system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/copyright.
```

```
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent  
permitted by applicable law.
```

```
Last login: Wed Jul 3 15:17:19 2019 from 192.168.1.42
```

```
SSH is enabled and the default password for the 'pi' user has not been changed.  
This is a security risk - please login as the 'pi' user and type 'passwd' to set a  
new password.
```

```
pi@raspberrypi:~ $ df -h
```

ファイルシステム	サイズ	使用	残り	使用%	マウント位置
/dev/root	3.4G	1.2G	2.1G	36%	/
devtmpfs	484M	0	484M	0%	/dev
tmpfs	488M	0	488M	0%	/dev/shm
tmpfs	488M	13M	476M	3%	/run
tmpfs	5.0M	4.0K	5.0M	1%	/run/lock
tmpfs	488M	0	488M	0%	/sys/fs/cgroup
/dev/mmcblk0p1	253M	40M	213M	16%	/boot
tmpfs	98M	0	98M	0%	/run/user/1000

```
pi@raspberrypi:~ $
```

```
pi@raspberrypi:~ $ cat /etc/apt/sources.list
```

```
deb http://raspbian.raspberrypi.org/raspbian/ buster main contrib non-free rpi  
# Uncomment line below then 'apt-get update' to enable 'apt-get source'  
#deb-src http://raspbian.raspberrypi.org/raspbian/ buster main contrib non-free rpi  
pi@raspberrypi:~ $
```

コンソール画面スナップショット

```

192.168.31.39 - Poderosa
ファイル(E)  編集(E)  コンソール(Q)  ツール(T)  ウィンドウ(W)  プラグイン(P)  ヘルプ(H)
SFTP SCP  改行 LF  インコーディング utf-8 (cjk)  generic
192.168.31.39
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed Jul  3 15:17:19 2019 from 192.168.1.42

SSH is enabled and the default password for the 'pi' user has not been changed.
This is a security risk - please login as the 'pi' user and type 'passwd' to set a new password.

pi@raspberrypi:~$ df -h
ファイルシステム サイズ 使用 残り 使用% マウント位置
/dev/root          3.4G  1.2G  2.1G   36% /
devtmpfs           484M    0  484M    0% /dev
tmpfs              488M    0  488M    0% /dev/shm
tmpfs              488M   13M  476M    3% /run
tmpfs              5.0M   4.0K   5.0M    1% /run/lock
tmpfs              488M    0  488M    0% /sys/fs/cgroup
/dev/mmcblk0p1    253M   40M  213M   16% /boot
tmpfs              98M    0   98M    0% /run/user/1000
pi@raspberrypi:~$ cat /etc/apt/sources.list
deb http://raspbian.raspberrypi.org/raspbian/ buster main contrib non-free rpi
# Uncomment line below then 'apt-get update' to enable 'apt-get source'
#deb-src http://raspbian.raspberrypi.org/raspbian/ buster main contrib non-free rpi
pi@raspberrypi:~$

```

2. システムの更新

apt-get update

```

pi@raspberrypi:~$ sudo apt-get update
取得:1 http://raspbian.raspberrypi.org/raspbian buster InRelease [15.0 kB]
取得:2 http://raspbian.raspberrypi.org/raspbian buster/main armhf Packages [13.0
MB]
取得:3 http://archive.raspberrypi.org/debian buster InRelease [25.1 kB]
取得:4 http://archive.raspberrypi.org/debian buster/main armhf Packages [204 kB]
取得:5 http://raspbian.raspberrypi.org/raspbian buster/contrib armhf Packages [58.7
kB]
13.3 MB を 1分 16秒 で取得しました (174 kB/s)
パッケージリストを読み込んでいます... 完了
pi@raspberrypi:~$

```

apt-get upgrade

```

pi@raspberrypi:~$ sudo apt-get upgrade
パッケージリストを読み込んでいます... 完了
依存関係ツリーを作成しています
状態情報を読み取っています... 完了
アップグレードパッケージを検出しています... 完了
以下のパッケージはアップグレードされます:
  bind9-host bzip2 cron dmsetup libbind9-161 libbz2-1.0 libdevmapper1.02.1 libdns-
export1104 libdns1104
  libexpat1 libgssapi-krb5-2 libisc-export1100 libisc1100 libisccc161 libisccfg163
  libk5crypto3 libkrb5-3
  libkrb5support0 liblwres161 libraspberrypi-bin libraspberrypi-dev libraspberrypi-
doc libraspberrypi0
  libwbclient0 nano raspberrypi-bootloader raspberrypi-kernel raspi-config
アップグレード: 28 個、新規インストール: 0 個、削除: 0 個、保留: 0 個。
97.9 MB のアーカイブを取得する必要があります。
この操作後に追加で 4,096 B のディスク容量が消費されます。
続行しますか? [Y/n] y
取得:1 http://archive.raspberrypi.org/debian buster/main armhf libraspberrypi-doc

```

```
armhf 1.20190620+1-1 [31.4 MB]
取得:2 http://ftp.jaist.ac.jp/pub/Linux/raspbian-archive/raspbian buster/main armhf
cron armhf 3.0p11-134 [93.8 kB]
```

途中省略

```
libdevmapper1.02.1:armhf (2:1.02.155-3) を設定しています ...
dmsetup (2:1.02.155-3) を設定しています ...
install-info (6.5.0.dfsg.1-4+b1) のトリガを処理しています ...
libc-bin (2.28-10+rpi1) のトリガを処理しています ...
systemd (241-5+rpi1) のトリガを処理しています ...
man-db (2.8.5-2) のトリガを処理しています ...
pi@raspberrypi:~ $
```

1. vim を追加

```
pi@raspberrypi:~ $ sudo apt-get -y install vim
パッケージリストを読み込んでいます... 完了
依存関係ツリーを作成しています
状態情報を読み取っています... 完了
以下の追加パッケージがインストールされます:
  vim-runtime
提案パッケージ:
  ctags vim-doc vim-scripts
以下のパッケージが新たにインストールされます:
  vim vim-runtime
アップグレード: 0 個、新規インストール: 2 個、削除: 0 個、保留: 0 個。
6,780 kB のアーカイブを取得する必要があります。
この操作後に追加で 32.6 MB のディスク容量が消費されます。
取得:1 http://ftp.jaist.ac.jp/pub/Linux/raspbian-archive/raspbian buster/main armhf
vim-runtime all 2:8.1.0875-5 [5,775 kB]
取得:2 http://ftp.jaist.ac.jp/pub/Linux/raspbian-archive/raspbian buster/main armhf
vim armhf 2:8.1.0875-5 [1,005 kB]
6,780 kB を 13 秒 で取得しました (517 kB/s)
以前に未選択のパッケージ vim-runtime を選択しています。
(データベースを読み込んでいます ... 現在 37607 個のファイルとディレクトリがインスト
ールされています。)
```

```
.../vim-runtime_2%3a8.1.0875-5_all.deb を展開する準備をしています ...
'vim-runtime による /usr/share/vim/vim81/doc/help.txt から
/usr/share/vim/vim81/doc/help.txt.vim-tiny への退避 (divert)' を追加しています
'vim-runtime による /usr/share/vim/vim81/doc/tags から
/usr/share/vim/vim81/doc/tags.vim-tiny への退避 (divert)' を追加しています
vim-runtime (2:8.1.0875-5) を展開しています...
以前に未選択のパッケージ vim を選択しています。
.../vim_2%3a8.1.0875-5_armhf.deb を展開する準備をしています ...
vim (2:8.1.0875-5) を展開しています...
vim-runtime (2:8.1.0875-5) を設定しています ...
vim (2:8.1.0875-5) を設定しています ...
update-alternatives: /usr/bin/vim (vim) を提供するために自動モードで
/usr/bin/vim.basic を使います
update-alternatives: /usr/bin/vimdiff (vimdiff) を提供するために自動モードで
/usr/bin/vim.basic を使います
update-alternatives: /usr/bin/rvim (rvim) を提供するために自動モードで
/usr/bin/vim.basic を使います
update-alternatives: /usr/bin/rview (rview) を提供するために自動モードで
```

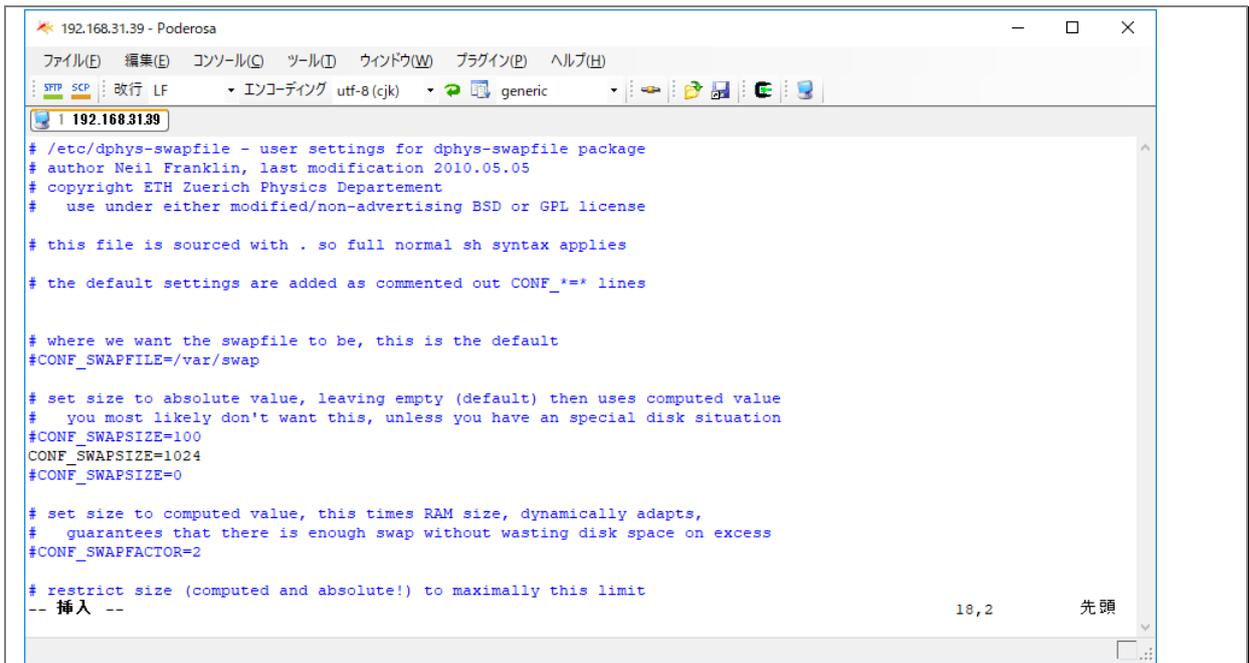
```

/usr/bin/vim.basic を使います
update-alternatives: /usr/bin/vi (vi) を提供するために自動モードで
/usr/bin/vim.basic を使います
update-alternatives: /usr/bin/view (view) を提供するために自動モードで
/usr/bin/vim.basic を使います
update-alternatives: /usr/bin/ex (ex) を提供するために自動モードで
/usr/bin/vim.basic を使います
man-db (2.8.5-2) のトリガを処理しています ...
pi@raspberrypi:~ $

```

1. SWAP ファイルを 1024KB に設定

後に `passenger` をインストールするため `swap` を拡張する



```

# /etc/dphys-swapfile - user settings for dphys-swapfile package
# author Neil Franklin, last modification 2010.05.05
# copyright ETH Zuerich Physics Departement
# use under either modified/non-advertising BSD or GPL license

# this file is sourced with . so full normal sh syntax applies

# the default settings are added as commented out CONF_*=* lines

# where we want the swapfile to be, this is the default
#CONF_SWAPFILE=/var/swap

# set size to absolute value, leaving empty (default) then uses computed value
# you most likely don't want this, unless you have an special disk situation
#CONF_SWAPSIZE=100
CONF_SWAPSIZE=1024
#CONF_SWAPSIZE=0

# set size to computed value, this times RAM size, dynamically adapts,
# guarantees that there is enough swap without wasting disk space on excess
#CONF_SWAPFACTOR=2

# restrict size (computed and absolute!) to maximally this limit
-- 挿入 --

```

`/etc/dphys-swapfile` を修正する。

```

pi@raspberrypi:~ $ sudo vi /etc/dphys-swapfile
#CONF_SWAPSIZE=100
CONF_SWAPSIZE=1024
#CONF_SWAPSIZE=0

```

```

pi@raspberrypi:~ $ sudo dphys-swapfile install
want /var/swap=1024MByte, checking existing: deleting wrong size file (104857600),
generating swapfile ... of 1024MBytes
pi@raspberrypi:~ $ sudo dphys-swapfile swapon
pi@raspberrypi:~ $ free

```

	total	used	free	shared	buff/cache	available
Mem:	999036	54736	228840	12868	715460	865988
Swap:	1048572	0	1048572			

```

pi@raspberrypi:~ $

```

拡張後のメモリ状態確認

```

192.168.31.39 - Poderosa
ファイル(F) 編集(E)  コンソール(C) ツール(T) ウィンドウ(W) プラグイン(P) ヘルプ(H)
SFTP SCP 改行 LF  インコーディング utf-8 (cjk)  generic
192.168.31.39
を追加しています
vim-runtime (2:8.1.0875-5) を展開しています...
以前に未選択のパッケージ vim を選択しています。
.../vim_2%3a8.1.0875-5_armhf.deb を展開する準備をしています...
vim (2:8.1.0875-5) を展開しています...
vim-runtime (2:8.1.0875-5) を設定しています...
vim (2:8.1.0875-5) を設定しています...
update-alternatives: /usr/bin/vim (vim) を提供するために自動モードで /usr/bin/vim.basic を使います
update-alternatives: /usr/bin/vimdiff (vimdiff) を提供するために自動モードで /usr/bin/vim.basic を使います
update-alternatives: /usr/bin/rvim (rvim) を提供するために自動モードで /usr/bin/vim.basic を使います
update-alternatives: /usr/bin/rview (rview) を提供するために自動モードで /usr/bin/vim.basic を使います
update-alternatives: /usr/bin/vi (vi) を提供するために自動モードで /usr/bin/vim.basic を使います
update-alternatives: /usr/bin/view (view) を提供するために自動モードで /usr/bin/vim.basic を使います
update-alternatives: /usr/bin/ex (ex) を提供するために自動モードで /usr/bin/vim.basic を使います
man-db (2.8.5-2) のトリガを処理しています...
pi@raspberrypi:~$ sudo vi /etc/dphys-swapfile
pi@raspberrypi:~$ sudo dphys-swapfile install
want /var/swap=1024MByte, checking existing: deleting wrong size file (104857600), generating swapfile ... of 1024
MBytes
pi@raspberrypi:~$ sudo dphys-swapfile swapon
pi@raspberrypi:~$ free
              total        used         free      shared  buff/cache   available
Mem:           999036         54736        228840        12868        715460        865988
Swap:          1048572             0         1048572
pi@raspberrypi:~$

```

2. 一般ユーザーrmenu_userを作成する、

adduser を使うこと 「useradd」を使うとはまる、環境変数が引き継がれないので、諸設定が必要になるうえエディタも使いにくい

rmenu_user

セキュリティポリシーに基づいて任意のパスワードを設定します。

```

pi@raspberrypi:~$ sudo adduser rmenu_user
ユーザ `rmenu_user' を追加しています...
新しいグループ `rmenu_user' (1001) を追加しています...
新しいユーザ `rmenu_user' (1001) をグループ `rmenu_user' として追加しています...
ホームディレクトリ `/home/rmenu_user' を作成しています...
`/etc/skel' からファイルをコピーしています...
新しいパスワード:
新しいパスワードを再入力してください:
passwd: パスワードは正しく更新されました
rmenu_user のユーザ情報を変更中
新しい値を入力してください。標準設定値を使うならリターンを押してください
フルネーム []:
部屋番号 []:
職場電話番号 []:
自宅電話番号 []:
その他 []:
以上で正しいですか? [Y/n] y
pi@raspberrypi:~$

```

```
192.168.31.39 - Poderosa
ファイル(E) 編集(E)  コンソール(Q) ツール(T) ウィンドウ(W) プラグイン(P) ヘルプ(H)
SFTP SCP  改行 LF  インコーディング utf-8 (cjk)  generic
192.168.31.39
pi@raspberrypi:~$ sudo vi /etc/dphys-swapfile
pi@raspberrypi:~$ sudo dphys-swapfile install
want /var/swap=1024MByte, checking existing: deleting wrong size file (104857600), generating swapfile ... of 1024
MBytes
pi@raspberrypi:~$ sudo dphys-swapfile swapon
pi@raspberrypi:~$ free
              total        used         free      shared  buff/cache   available
Mem:           999036          54736       228840        12868        715460       865988
Swap:        1048572              0       1048572
pi@raspberrypi:~$ sudo adduser rmenu user
ユーザ `rmenu user' を追加しています...
新しいグループ `rmenu user' (1001) を追加しています...
新しいユーザ `rmenu user' (1001) をグループ `rmenu user' として追加しています...
ホームディレクトリ `/home/rmenu user' を作成しています...
`/etc/skel' からファイルをコピーしています...
新しいパスワード:
新しいパスワードを再入力してください:
passwd: パスワードは正しく更新されました
rmenu user のユーザ情報を変更中
新しい値を入力してください。標準設定値を使うならリターンを押してください
フルネーム []:
部屋番号 []:
職場電話番号 []:
自宅電話番号 []:
その他 []:
以上で正しいですか? [Y/n] y
pi@raspberrypi:~$
```

第6章 apt-get パッケージ管理

1. readline-devel flex bison zlib-devel を追加

libreadline-dev

```
pi@raspberrypi:~ $ sudo apt-get -y install libreadline-dev
パッケージリストを読み込んでいます... 完了
依存関係ツリーを作成しています
状態情報を読み取っています... 完了
以下の追加パッケージがインストールされます:
  libncurses-dev
提案パッケージ:
  ncurses-doc readline-doc
以下のパッケージが新たにインストールされます:
  libncurses-dev libreadline-dev
アップグレード: 0 個、新規インストール: 2 個、削除: 0 個、保留: 0 個。
388 kB のアーカイブを取得する必要があります。
この操作後に追加で 2,189 kB のディスク容量が消費されます。
取得:1 http://ftp.jaist.ac.jp/pub/Linux/raspbian-archive/raspbian buster/main armhf
libncurses-dev armhf 6.1+20181013-2 [278 kB]
取得:2 http://ftp.jaist.ac.jp/pub/Linux/raspbian-archive/raspbian buster/main armhf
libreadline-dev armhf 7.0-5 [110 kB]
388 kB を 2 秒 で取得しました (177 kB/s)
以前に未選択のパッケージ libncurses-dev:armhf を選択しています。
(データベースを読み込んでいます ... 現在 39414 個のファイルとディレクトリがインストールされています。)
.../libncurses-dev_6.1+20181013-2_armhf.deb を展開する準備をしています ...
libncurses-dev:armhf (6.1+20181013-2) を展開しています...
以前に未選択のパッケージ libreadline-dev:armhf を選択しています。
.../libreadline-dev_7.0-5_armhf.deb を展開する準備をしています ...
libreadline-dev:armhf (7.0-5) を展開しています...
libncurses-dev:armhf (6.1+20181013-2) を設定しています ...
libreadline-dev:armhf (7.0-5) を設定しています ...
install-info (6.5.0.dfsg.1-4+b1) のトリガを処理しています ...
man-db (2.8.5-2) のトリガを処理しています ...
pi@raspberrypi:~ $
```

libreadline-dev

```
pi@raspberrypi:~ $ sudo apt-get -y install libreadline-dev
パッケージリストを読み込んでいます... 完了
依存関係ツリーを作成しています
状態情報を読み取っています... 完了
libreadline-dev はすでに最新バージョン (7.0-5) です。
アップグレード: 0 個、新規インストール: 0 個、削除: 0 個、保留: 0 個。
pi@raspberrypi:~ $
```

flex、bison

```
pi@raspberrypi:~ $ sudo apt-get -y install flex bison
パッケージリストを読み込んでいます... 完了
依存関係ツリーを作成しています
状態情報を読み取っています... 完了
以下の追加パッケージがインストールされます:
  libbison-dev libfl-dev libfl2 libsigsegv2 m4
提案パッケージ:
```

```

bison-doc flex-doc m4-doc
以下のパッケージが新たにインストールされます:
  bison flex libbison-dev libfl-dev libfl2 libsigsegv2 m4
アップグレード: 0 個、新規インストール: 7 個、削除: 0 個、保留: 0 個。
2,180 kB のアーカイブを取得する必要があります。
この操作後に追加で 4,502 kB のディスク容量が消費されます。
取得:1 http://ftp.jaist.ac.jp/pub/Linux/raspbian-archive/raspbian buster/main armhf
libsigsegv2 armhf 2.12-2 [32.3 kB]
取得:2 http://ftp.jaist.ac.jp/pub/Linux/raspbian-archive/raspbian buster/main armhf
m4 armhf 1.4.18-2 [185 kB]
取得:3 http://ftp.jaist.ac.jp/pub/Linux/raspbian-archive/raspbian buster/main armhf
flex armhf 2.6.4-6.2 [427 kB]
取得:4 http://ftp.jaist.ac.jp/pub/Linux/raspbian-archive/raspbian buster/main armhf
libbison-dev armhf 2:3.3.2.dfsg-1 [500 kB]
取得:5 http://ftp.jaist.ac.jp/pub/Linux/raspbian-archive/raspbian buster/main armhf
bison armhf 2:3.3.2.dfsg-1 [829 kB]
取得:6 http://ftp.jaist.ac.jp/pub/Linux/raspbian-archive/raspbian buster/main armhf
libfl2 armhf 2.6.4-6.2 [103 kB]
取得:7 http://ftp.jaist.ac.jp/pub/Linux/raspbian-archive/raspbian buster/main armhf
libfl-dev armhf 2.6.4-6.2 [104 kB]
2,180 kB を 6 秒 で取得しました (358 kB/s)
以前に未選択のパッケージ libsigsegv2:armhf を選択しています。
(データベースを読み込んでいます ... 現在 39517 個のファイルとディレクトリがインスト
ールされています。)
```

```

.../0-libsigsegv2_2.12-2_armhf.deb を展開する準備をしています ...
libsigsegv2:armhf (2.12-2) を展開しています...
以前に未選択のパッケージ m4 を選択しています。
.../1-m4_1.4.18-2_armhf.deb を展開する準備をしています ...
m4 (1.4.18-2) を展開しています...
以前に未選択のパッケージ flex を選択しています。
.../2-flex_2.6.4-6.2_armhf.deb を展開する準備をしています ...
flex (2.6.4-6.2) を展開しています...
以前に未選択のパッケージ libbison-dev:armhf を選択しています。
.../3-libbison-dev_2%3a3.3.2.dfsg-1_armhf.deb を展開する準備をしています ...
libbison-dev:armhf (2:3.3.2.dfsg-1) を展開しています...
以前に未選択のパッケージ bison を選択しています。
.../4-bison_2%3a3.3.2.dfsg-1_armhf.deb を展開する準備をしています ...
bison (2:3.3.2.dfsg-1) を展開しています...
以前に未選択のパッケージ libfl2:armhf を選択しています。
.../5-libfl2_2.6.4-6.2_armhf.deb を展開する準備をしています ...
libfl2:armhf (2.6.4-6.2) を展開しています...
以前に未選択のパッケージ libfl-dev:armhf を選択しています。
.../6-libfl-dev_2.6.4-6.2_armhf.deb を展開する準備をしています ...
libfl-dev:armhf (2.6.4-6.2) を展開しています...
libbison-dev:armhf (2:3.3.2.dfsg-1) を設定しています ...
libsigsegv2:armhf (2.12-2) を設定しています ...
libfl2:armhf (2.6.4-6.2) を設定しています ...
m4 (1.4.18-2) を設定しています ...
bison (2:3.3.2.dfsg-1) を設定しています ...
update-alternatives: /usr/bin/yacc (yacc) を提供するために自動モードで
/usr/bin/bison.yacc を使います
flex (2.6.4-6.2) を設定しています ...
libfl-dev:armhf (2.6.4-6.2) を設定しています ...
libc-bin (2.28-10+rpi1) のトリガを処理しています ...
man-db (2.8.5-2) のトリガを処理しています ...

```

```
install-info (6.5.0.dfsg.1-4+b1) のトリガを処理しています ...
pi@raspberrypi:~ $
```

libxslt1-dev

```
pi@raspberrypi:~ $ sudo apt-get -y install libxslt1-dev
パッケージリストを読み込んでいます... 完了
依存関係ツリーを作成しています
状態情報を読み取っています... 完了
以下の追加パッケージがインストールされます:
  icu-devtools libicu-dev libxml2-dev libxslt1.1
提案パッケージ:
  icu-doc
以下のパッケージが新たにインストールされます:
  icu-devtools libicu-dev libxml2-dev libxslt1-dev libxslt1.1
アップグレード: 0 個、新規インストール: 5 個、削除: 0 個、保留: 0 個。
10.3 MB のアーカイブを取得する必要があります。
この操作後に追加で 45.7 MB のディスク容量が消費されます。
取得:1 http://ftp.jaist.ac.jp/pub/Linux/raspbian-archive/raspbian buster/main armhf
icu-devtools armhf 63.1-6 [163 kB]
取得:2 http://ftp.jaist.ac.jp/pub/Linux/raspbian-archive/raspbian buster/main armhf
libicu-dev armhf 63.1-6 [8,841 kB]
取得:3 http://ftp.jaist.ac.jp/pub/Linux/raspbian-archive/raspbian buster/main armhf
libxml2-dev armhf 2.9.4+dfsg1-7+b1 [640 kB]
取得:4 http://ftp.jaist.ac.jp/pub/Linux/raspbian-archive/raspbian buster/main armhf
libxslt1.1 armhf 1.1.32-2 [215 kB]
取得:5 http://ftp.jaist.ac.jp/pub/Linux/raspbian-archive/raspbian buster/main armhf
libxslt1-dev armhf 1.1.32-2 [492 kB]
10.3 MB を 24 秒 で取得しました (436 kB/s)
以前に未選択のパッケージ icu-devtools を選択しています。
(データベースを読み込んでいます ... 現在 39830 個のファイルとディレクトリがインストールされています。)
```

```
.../icu-devtools_63.1-6_armhf.deb を展開する準備をしています ...
icu-devtools (63.1-6) を展開しています...
以前に未選択のパッケージ libicu-dev:armhf を選択しています。
.../libicu-dev_63.1-6_armhf.deb を展開する準備をしています ...
libicu-dev:armhf (63.1-6) を展開しています...
以前に未選択のパッケージ libxml2-dev:armhf を選択しています。
.../libxml2-dev_2.9.4+dfsg1-7+b1_armhf.deb を展開する準備をしています ...
libxml2-dev:armhf (2.9.4+dfsg1-7+b1) を展開しています...
以前に未選択のパッケージ libxslt1.1:armhf を選択しています。
.../libxslt1.1_1.1.32-2_armhf.deb を展開する準備をしています ...
libxslt1.1:armhf (1.1.32-2) を展開しています...
以前に未選択のパッケージ libxslt1-dev:armhf を選択しています。
.../libxslt1-dev_1.1.32-2_armhf.deb を展開する準備をしています ...
libxslt1-dev:armhf (1.1.32-2) を展開しています...
icu-devtools (63.1-6) を設定しています ...
libxslt1.1:armhf (1.1.32-2) を設定しています ...
libicu-dev:armhf (63.1-6) を設定しています ...
libxml2-dev:armhf (2.9.4+dfsg1-7+b1) を設定しています ...
libxslt1-dev:armhf (1.1.32-2) を設定しています ...
man-db (2.8.5-2) のトリガを処理しています ...
libc-bin (2.28-10+rpi1) のトリガを処理しています ...
pi@raspberrypi:~ $
```

2. 日本語処理に必要な nkf コマンドをインストールする

```

pi@raspberrypi:~$ sudo apt-get -y install nkf
パッケージリストを読み込んでいます... 完了
依存関係ツリーを作成しています
状態情報を読み取っています... 完了
以下のパッケージが新たにインストールされます:
  nkf
アップグレード: 0 個、新規インストール: 1 個、削除: 0 個、保留: 0 個。
129 kB のアーカイブを取得する必要があります。
この操作後に追加で 294 kB のディスク容量が消費されます。
取得:1 http://ftp.jaist.ac.jp/pub/Linux/raspbian-archive/raspbian buster/main armhf
nkf armhf 1:2.1.5-1 [129 kB]
129 kB を 2 秒 で取得しました (74.1 kB/s)
以前に未選択のパッケージ nkf を選択しています。
(データベースを読み込んでいます ... 現在 46342 個のファイルとディレクトリがインスト
ールされています。)
.../nkf_1%3a2.1.5-1_armhf.deb を展開する準備をしています ...
nkf (1:2.1.5-1) を展開しています...
nkf (1:2.1.5-1) を設定しています ...
man-db (2.8.5-2) のトリガを処理しています ...
pi@raspberrypi:~$

```

3. tcsh をインストールする

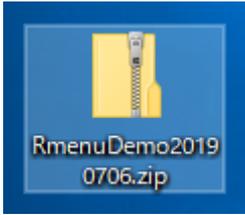
```

pi@raspberrypi:~$ sudo apt-get -y install tcsh
パッケージリストを読み込んでいます... 完了
依存関係ツリーを作成しています
状態情報を読み取っています... 完了
以下のパッケージが新たにインストールされます:
  tcsh
アップグレード: 0 個、新規インストール: 1 個、削除: 0 個、保留: 0 個。
428 kB のアーカイブを取得する必要があります。
この操作後に追加で 1,249 kB のディスク容量が消費されます。
取得:1 http://ftp.jaist.ac.jp/pub/Linux/raspbian-archive/raspbian buster/main armhf
tcsh armhf 6.20.00-7 [428 kB]
428 kB を 1 秒 で取得しました (382 kB/s)
以前に未選択のパッケージ tcsh を選択しています。
(データベースを読み込んでいます ... 現在 46351 個のファイルとディレクトリがインスト
ールされています。)
.../tcsh_6.20.00-7_armhf.deb を展開する準備をしています ...
tcsh (6.20.00-7) を展開しています...
tcsh (6.20.00-7) を設定しています ...
update-alternatives: /bin/csh (csh) を提供するために自動モードで /bin/tcsh を使いま
す
man-db (2.8.5-2) のトリガを処理しています ...
pi@raspberrypi:~$

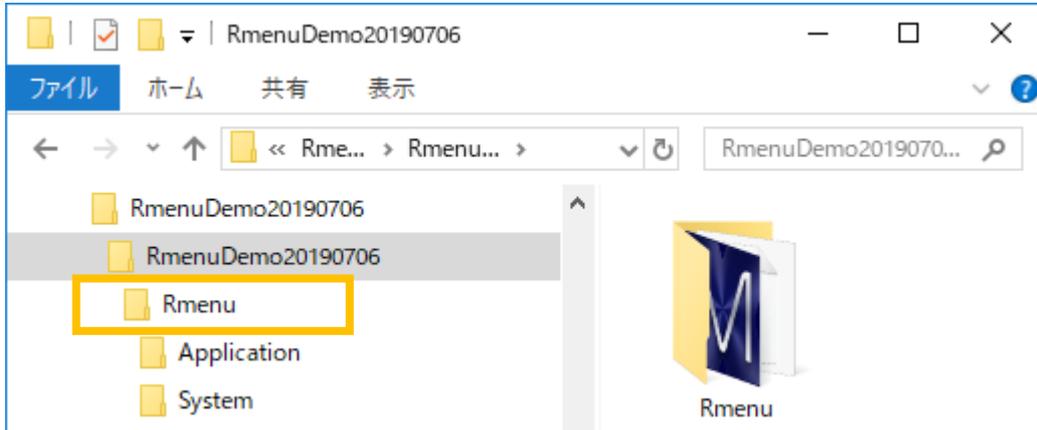
```

4. Rmenu フォルダを RaspberryPI の rmenu_user ディレクトリに置く

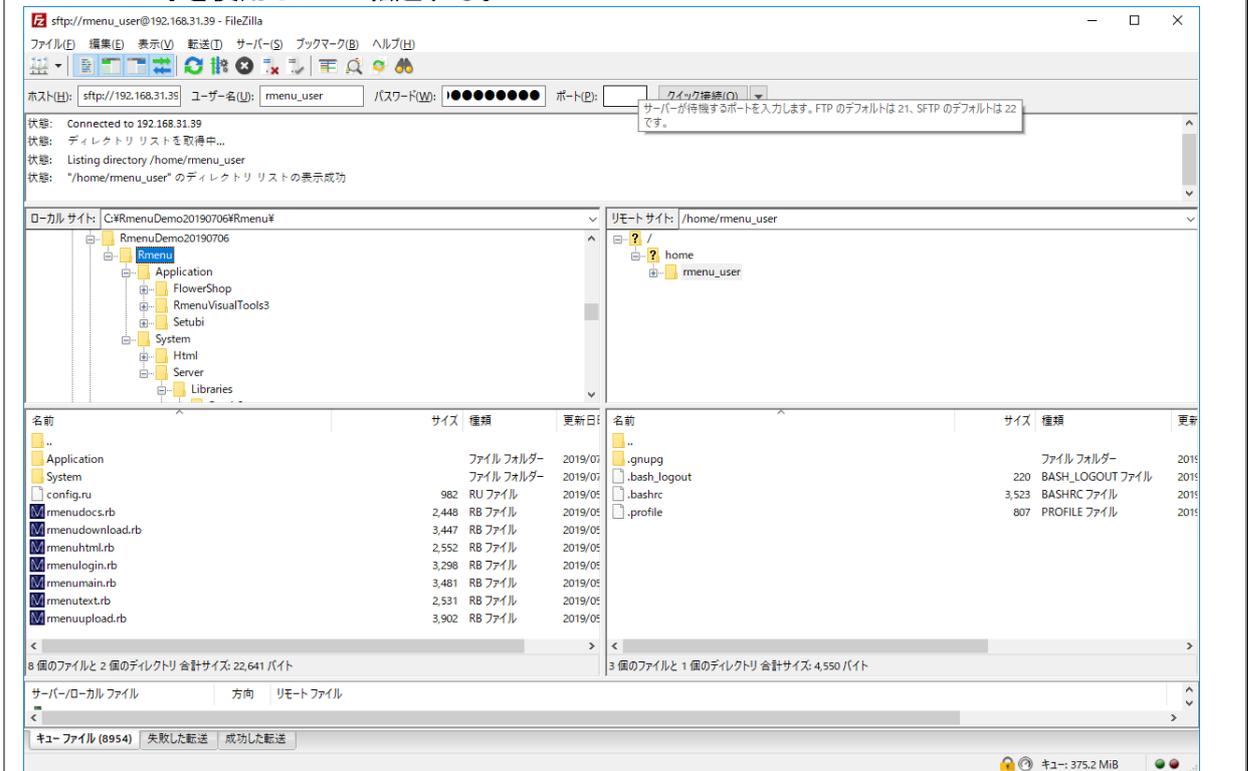
Rmenu システム一式 RmenuDemo20190706.zip を解凍し、



解凍状況



Rmenu フォルダを RaspberryPI の rmenu_user ディレクトリに転送してください。
FileZilla 等を使用し SFTP 転送する。



転送完了後

Host: sftp://192.168.31.39 User: rmenu_user Password: [REDACTED] Port: [REDACTED]

Status: Listing directory /home/mmenu_user
 Status: "/home/mmenu_user" のディレクトリリストの表示成功
 Status: サーバーから切断されました
 Status: サーバーから切断されました

名前	サイズ	種類	更新日時	名前	サイズ	種類	更新日時	パーミッション	所有者/グループ
Application		ディレクトリ	2019/07/02	home		ディレクトリ			
System		ディレクトリ	2019/07/02	rmenu		ディレクトリ	2019/07/05 15:59:38	drwxr-xr-x	rmenu_user rmenu_user
config.ru	982	RUファイル	2019/05/29	.bashrc	3,523	BASHRCファイル	2019/07/03 15:47:38	-rw-r--r--	rmenu_user rmenu_user
rmenudocs.rb	2,448	RBファイル	2019/05/29	.profile	807	PROFILEファイル	2019/07/03 15:47:38	-rw-r--r--	rmenu_user rmenu_user
rmenudownload.rb	3,447	RBファイル	2019/05/29						
rmenuhtml.rb	2,552	RBファイル	2019/05/29						
rmenulogin.rb	3,298	RBファイル	2019/05/29						
rmenumain.rb	3,481	RBファイル	2019/05/29						
rmenutext.rb	2,531	RBファイル	2019/05/29						
rmenuupload.rb	3,902	RBファイル	2019/05/29						

8 個のファイルと 2 個のディレクトリ 合計サイズ: 22,641 バイト

1 個のディレクトリを選択

キュー: ファイル (8954) | 失敗した転送 (2) | 成功した転送 (3840)

キュー: 375.2 MiB

Rmenu
RaspberryPi3

第7章 Apache2 の起動

1. 次の URL をブラウザで開く

http://ww.xx.yy.zz/



2. apache2 のインストール

```
pi@raspberrypi:~ $ sudo apt-get -y install apache2 apache2-dev
パッケージリストを読み込んでいます... 完了
依存関係ツリーを作成しています
状態情報を読み取っています... 完了
以下の追加パッケージがインストールされます:
  apache2-bin apache2-data apache2-utils autoconf automake autopoint autotools-dev
  debhelper dh-autoreconf
  dh-strip-nondeterminism dwz gettext intltool-debian libapr1 libapr1-dev
  libaprutil1 libaprutil1-dbd-sqlite3
  libaprutil1-dev libaprutil1-ldap libarchive-cpio-perl libarchive-zip-perl
  libbrotli1 libcroco3 libexpat1-dev
  libfile-stripnondeterminism-perl libjansson4 libldap2-dev libltdl-dev libltdl7
  liblua5.2-0
  libmail-sendmail-perl libsctp-dev libsctp1 libsys-hostname-long-perl libtool po-
  debconf ssl-cert uuid-dev
提案パッケージ:
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom www-browser autoconf-
  archive gnu-standards
  autoconf-doc dh-make gettext-doc libasprintf-dev libgettextpo-dev libtool-doc
  lksctp-tools gfortran
  | fortran95-compiler gcj-jdk libmail-box-perl openssl-blacklist
以下のパッケージが新たにインストールされます:
  apache2 apache2-bin apache2-data apache2-dev apache2-utils autoconf automake
  autopoint autotools-dev debhelper
  dh-autoreconf dh-strip-nondeterminism dwz gettext intltool-debian libapr1
  libapr1-dev libaprutil1
  libaprutil1-dbd-sqlite3 libaprutil1-dev libaprutil1-ldap libarchive-cpio-perl
  libarchive-zip-perl libbrotli1
  libcroco3 libexpat1-dev libfile-stripnondeterminism-perl libjansson4 libldap2-dev
  libltdl-dev libltdl7
  liblua5.2-0 libmail-sendmail-perl libsctp-dev libsctp1 libsys-hostname-long-perl
  libtool po-debconf ssl-cert
```

```

uid-dev
アップグレード: 0 個、新規インストール: 40 個、削除: 0 個、保留: 0 個。
10.0 MB のアーカイブを取得する必要があります。
この操作後に追加で 41.1 MB のディスク容量が消費されます。

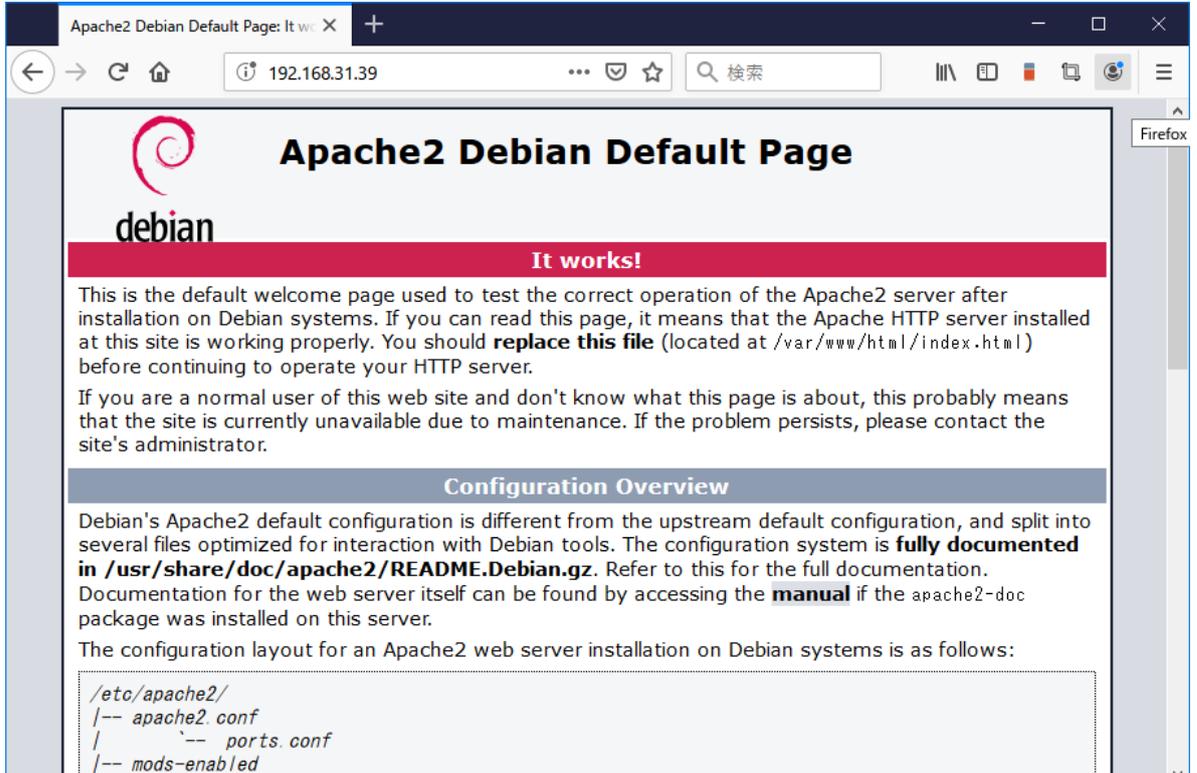
途中省略

che-htcacheclean.service.
debhelper (12.1.1) を設定しています ...
dh-autoreconf (19) を設定しています ...
dh-strip-nondeterminism (1.1.2-1) を設定しています ...
apache2-dev (2.4.38-3) を設定しています ...
libc-bin (2.28-10+rpi1) のトリガを処理しています ...
systemd (241-5+rpi1) のトリガを処理しています ...
man-db (2.8.5-2) のトリガを処理しています ...
install-info (6.5.0.dfsg.1-4+b1) のトリガを処理しています ...
pi@raspberrypi:~ $

```

3. http://ww.xx.yy.zz / を開く

正常に応答が帰るようになりました。



The screenshot shows a Firefox browser window with the address bar displaying '192.168.31.39'. The page content includes the Apache2 logo, the text 'It works!', and a 'Configuration Overview' section. The configuration overview text reads: 'Debian's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Debian tools. The configuration system is **fully documented** in **/usr/share/doc/apache2/README.Debian.gz**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server. The configuration layout for an Apache2 web server installation on Debian systems is as follows:'. Below this text is a code block showing the directory structure: `/etc/apache2/`, `-- apache2.conf`, `-- ports.conf`, `-- mods-enabled`.

4. アクセス権に www-data グループを追加

Rmenu ユーザディレクトリ rmenu_user の所有権・パーミッション設定

```
pi@raspberrypi:~ $ cd /home/  
pi@raspberrypi:/home $ ls -al  
合計 16  
drwxr-xr-x  4 root      root      4096  7月  3 15:47 .  
drwxr-xr-x 21 root      root      4096  7月  3 15:17 ..  
drwxr-xr-x  3 pi        pi        4096  7月  3 15:17 pi  
drwxr-xr-x  4 rmenu    user      4096  7月  3 15:59 rmenu_user  
pi@raspberrypi:/home $ sudo chown -R rmenu_user:www-data rmenu_user  
pi@raspberrypi:/home $ ls -al  
合計 16  
drwxr-xr-x  4 root      root      4096  7月  3 15:47 .  
drwxr-xr-x 21 root      root      4096  7月  3 15:17 ..  
drwxr-xr-x  3 pi        pi        4096  7月  3 15:17 pi  
drwxr-xr-x  4 rmenu_user www-data  4096  7月  3 15:59 rmenu_user  
pi@raspberrypi:/home $
```

この時点で次の URL へのアクセス権限が付与される。

http://ww.xx.yy.zz/rmenu_user/Application/Setubi/Html/Apps/Login/index.html

第8章 PostgreSQL インストール

1. postgresql インストール

デフォルトで postgresql11 がインストールされた。

```
pi@raspberrypi:~ $ sudo apt-get install postgresql
パッケージリストを読み込んでいます... 完了
依存関係ツリーを作成しています
状態情報を読み取っています... 完了
以下の追加パッケージがインストールされます:
 libllvm7 libpq5 libsensors-config libsensors5 postgresql-11 postgresql-client-11
 postgresql-client-common
 postgresql-common sysstat
提案パッケージ:
 lm-sensors postgresql-doc postgresql-doc-11 libjson-perl isag
以下のパッケージが新たにインストールされます:
 libllvm7 libpq5 libsensors-config libsensors5 postgresql postgresql-11
 postgresql-client-11
 postgresql-client-common postgresql-common sysstat
アップグレード: 0 個、新規インストール: 10 個、削除: 0 個、保留: 0 個。
26.9 MB のアーカイブを取得する必要があります。
この操作後に追加で 105 MB のディスク容量が消費されます。
続行しますか? [Y/n]
```

途中省略

```
debian found in ID_LIKE, treating as Debian
ユーザ postgres をグループ ssl-cert に追加
```

```
Creating config file /etc/postgresql-common/createcluster.conf with new version
Building PostgreSQL dictionaries from installed myspell/hunspell packages...
Removing obsolete dictionary files:
Created symlink /etc/systemd/system/multi-user.target.wants/postgresql.service →
 /lib/systemd/system/postgresql.service.
libsensors5:armhf (1:3.5.0-3) を設定しています ...
libllvm7:armhf (1:7.0.1-8+rpi1) を設定しています ...
sysstat (12.0.3-2) を設定しています ...
```

```
Creating config file /etc/default/sysstat with new version
update-alternatives: /usr/bin/sar (sar) を提供するために自動モードで
 /usr/bin/sar.sysstat を使います
Created symlink /etc/systemd/system/multi-user.target.wants/sysstat.service →
 /lib/systemd/system/sysstat.service.
postgresql-11 (11.4-1) を設定しています ...
Creating new PostgreSQL cluster 11/main ...
/usr/lib/postgresql/11/bin/initdb -D /var/lib/postgresql/11/main --auth-local peer
--auth-host md5
データベースシステム内のファイルの所有者は"postgres"となります。
このユーザがサーバプロセスも所有する必要があります。
```

```
データベースクラスタはロケール"ja_JP.UTF-8"で初期化されます。
そのためデフォルトのデータベース符号化方式はUTF8に設定されました。
initdb: ロケール"ja_JP.UTF-8"用の適切なテキスト検索設定が見つかりません
デフォルトのテキスト検索設定はsimpleに設定されました。
```

```
データベースのチェックサムは無効です。
```

```

既存のディレクトリ/var/lib/postgresql/11/mainの権限を修正します ... 完了
サブディレクトリを作成します ... 完了
max_connectionsのデフォルト値を選択します ... 100
shared_buffersのデフォルト値を選択します ... 128MB
selecting default timezone ... Asia/Tokyo
動的共有メモリの実装を選択します ... posix
設定ファイルを作成します ... 完了
ブートストラップスクリプトを実行します ... 完了
ブートストラップ後の初期化を行っています ... 完了
データをディスクに同期します...完了

```

成功しました。以下のようにしてデータベースサーバを起動できます。

```
pg_ctlcluster 11 main start
```

```

Ver Cluster Port Status Owner   Data directory          Log file
11 main    5432 down   postgres /var/lib/postgresql/11/main
/var/log/postgresql/postgresql-11-main.log
update-alternatives: /usr/share/man/man1/postmaster.1.gz (postmaster.1.gz) を提供す
るために自動モードで /usr/share/postgresql/11/man/man1/postmaster.1.gz を使います
postgresql (11+200+deb10u1) を設定しています ...
systemd (241-5+rpi1) のトリガを処理しています ...
man-db (2.8.5-2) のトリガを処理しています ...
libc-bin (2.28-10+rpi1) のトリガを処理しています ...
pi@raspberrypi:~ $

```

2. postgres ユーザーにパスワードを設定

```

pi@raspberrypi:~ $ sudo passwd postgres
新しい UNIX パスワードを入力してください:
新しい UNIX パスワードを再入力してください:
passwd: パスワードは正しく更新されました
pi@raspberrypi:~ $

```

3. TCP 接続の設定

`/usr/local/pgsql/data/pg_hba.conf` ファイルで認証設定を行います。

```

pi@raspberrypi:~ $ sudo vi /etc/postgresql/11/main/postgresql.conf
listen_addresses = '*'

```

アクセスを許可するには、次の行を加えます。

```
pi@raspberrypi:~ $ sudo vi /etc/postgresql/11/main/pg_hba.conf
```

```

host    all    all    127.0.0.1    trust
host    all    all    10.10.223.0/24    trust
#host   all    all    192.168.0.0/16    trust

```

PostgreSQL を再起動します。 `pg_hba.conf` の反映は `reload` で OK

```
pi@raspberrypi:~ $ sudo service postgresql restart
```

4. db をリストアする

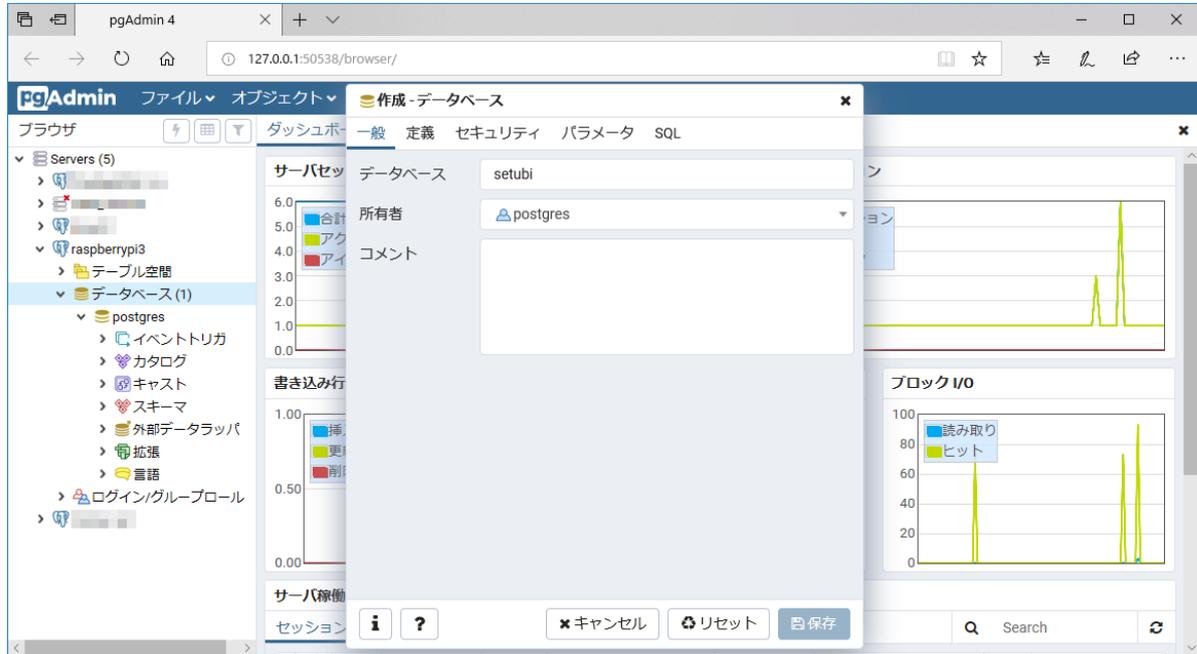
PgAdmin4 を使用し RaspberryPI に接続する。

次の 3 個の新規データベースを作成する。

flowershop

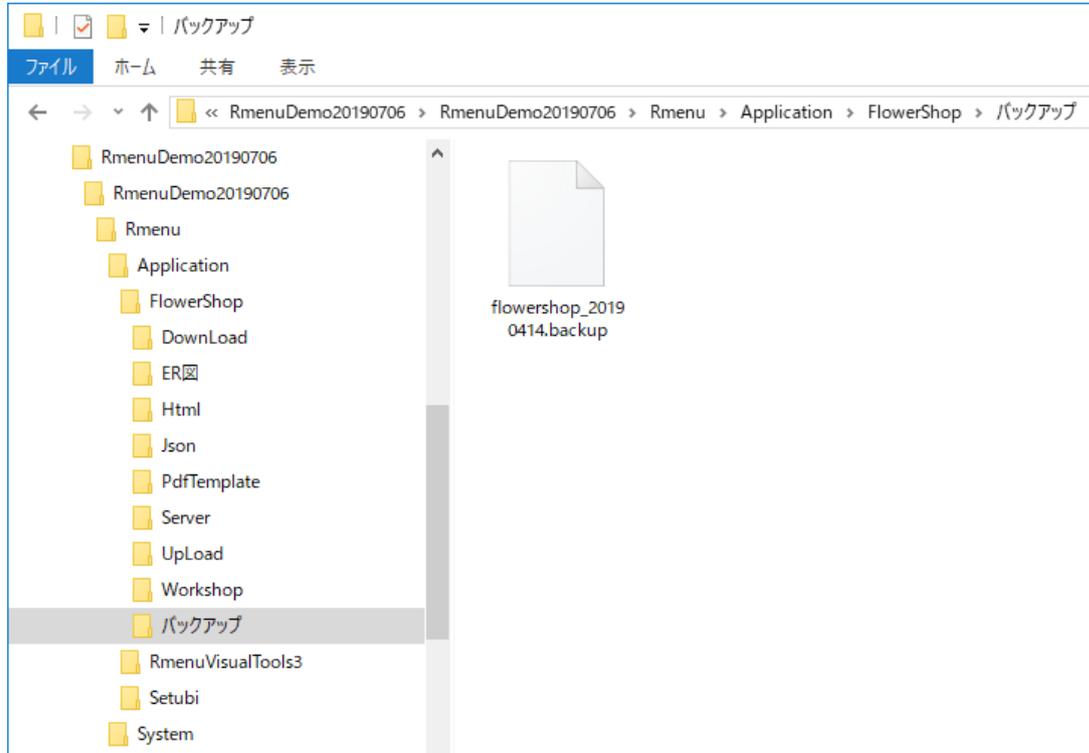
setubi

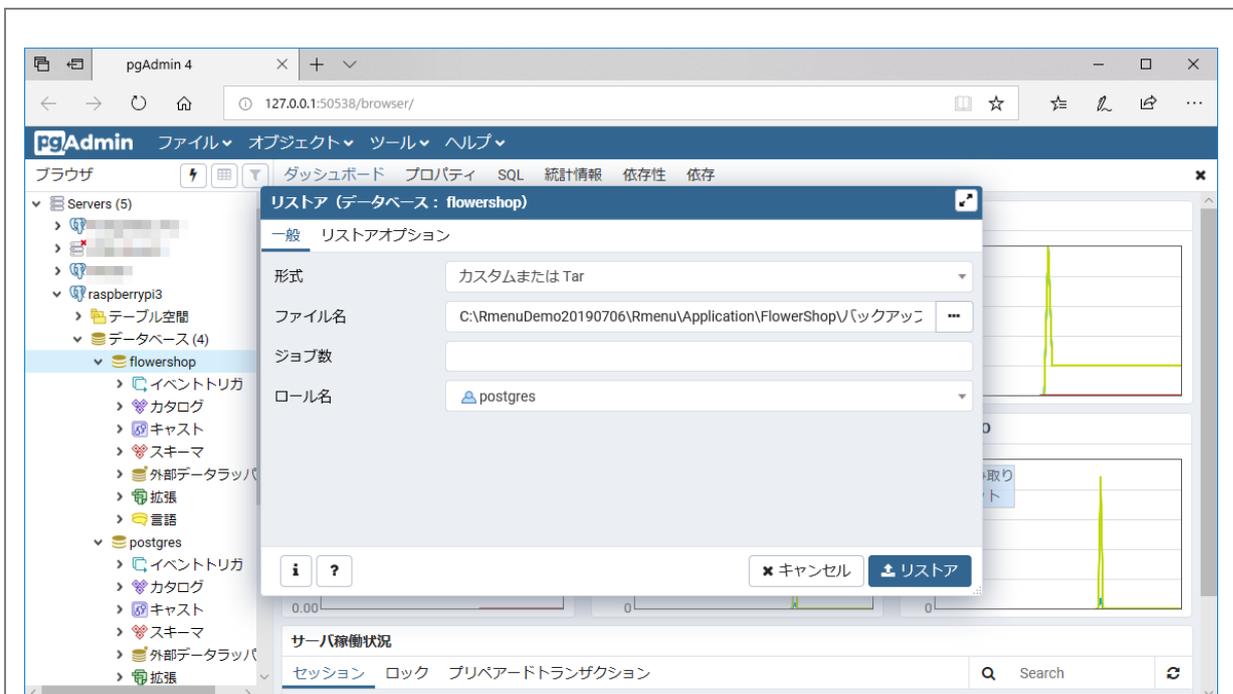
rmenuvisualtools3



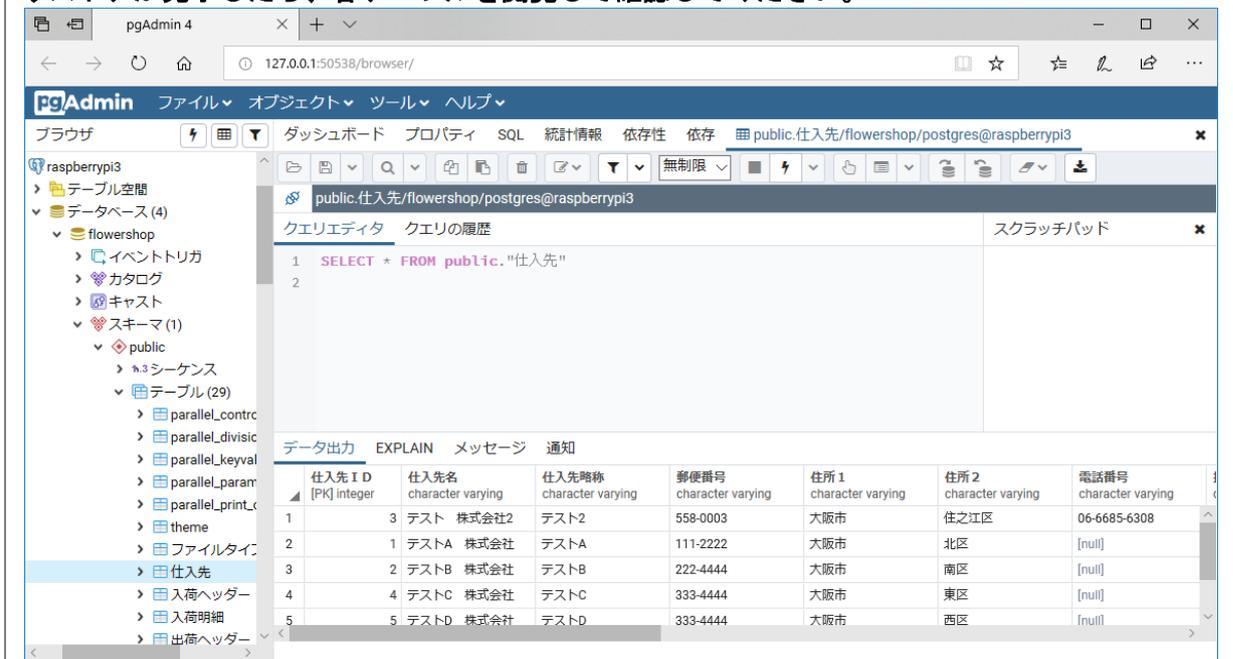
C:\RmenuDemo20190706\Rmenu\Application

解凍した RmenuDemo20190706 フォルダ の Application フォルダ内のアプリケーションないにバックアップフォルダがあります。それぞれ作成したデータベースにリストアしてください。





リストアが完了したら、各テーブルを閲覧して確認してください。



第9章 RVM および Ruby のインストール

1. RVM のインストール、(マルチユーザ) root でインストール

```
pi@raspberrypi:~ $ sudo su -

SSH is enabled and the default password for the 'pi' user has not been changed.
This is a security risk - please login as the 'pi' user and type 'passwd' to set a
new password.

root@raspberrypi:~# curl -L https://get.rvm.io | bash -s stable
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           % Dload  % Upload   Total   Spent    Left   Speed
100 194    100 194    0    0   248      0  --:--:-- --:--:-- --:--:--    247
100 24185  100 24185  0    0 26005      0  --:--:-- --:--:-- --:--:-- 26005
Downloading https://github.com/rvm/rvm/archive/1.29.8.tar.gz
Downloading https://github.com/rvm/rvm/releases/download/1.29.8/1.29.8.tar.gz.asc
gpg: ディレクトリ '/root/.gnupg' が作成されました
gpg: keybox '/root/.gnupg/pubring.kbx' が作成されました
gpg: 2019年05月08日 23時14分49秒 JST に施された署名
gpg:          RSA 鍵 7D2BAF1CF37B13E2069D6956105BD0E739499BDB を使用
gpg: 署名を検査できません: No public key
GPG signature verification failed for '/usr/local/rvm/archives/rvm-1.29.8.tgz' -
'https://github.com/rvm/rvm/releases/download/1.29.8/1.29.8.tar.gz.asc'! Try to
install GPG v2 and then fetch the public key:

    gpg --keyserver hkp://pool.sks-keyservers.net --recv-keys
409B6B1796C275462A1703113804BB82D39DC0E3 7D2BAF1CF37B13E2069D6956105BD0E739499BDB

or if it fails:

    command curl -sSL https://rvm.io/mpapis.asc | gpg --import -
    command curl -sSL https://rvm.io/pkuczynski.asc | gpg --import -

In case of further problems with validation please refer to
https://rvm.io/rvm/security

root@raspberrypi:~#
```

2. GPG signature を取得する

```
root@raspberrypi:~# curl -sSL https://rvm.io/mpapis.asc | gpg --import -
gpg: key 3804BB82D39DC0E3: 鍵がないため 47 個の署名は検査しません
gpg: /root/.gnupg/trustdb.gpg: 信用データベースができました
gpg: 鍵 3804BB82D39DC0E3: 公開鍵"Michal Papis (RVM signing) <mpapis@gmail.com>"をインポートしました
gpg:          処理数の合計: 1
gpg:          インポート: 1
gpg: 究極的に信用する鍵が見つかりません
root@raspberrypi:~# curl -sSL https://rvm.io/pkuczynski.asc | gpg --import -
gpg: 鍵 105BD0E739499BDB: 公開鍵"Piotr Kuczynski <piotr.kuczynski@gmail.com>"をインポートしました
gpg:          処理数の合計: 1
gpg:          インポート: 1
root@raspberrypi:~#
```

3. GPG signature を取得後、再実行

```
root@raspberrypi:~# curl -L https://get.rvm.io | bash -s stable
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           %             %             Dload  Upload  Total  Spent    Left   Speed
100  194    100  194    0     0    250      0  --:--:-- --:--:-- --:--:--    250
100 24185   100 24185    0     0 26345      0  --:--:-- --:--:-- --:--:-- 26345
Downloading https://github.com/rvm/rvm/archive/1.29.8.tar.gz
Downloading https://github.com/rvm/rvm/releases/download/1.29.8/1.29.8.tar.gz.asc
gpg: 2019 年 05 月 08 日 23 時 14 分 49 秒 JST に施された署名
gpg: RSA 鍵 7D2BAF1CF37B13E2069D6956105BD0E739499BDB を使用
gpg: "Piotr Kuczynski <piotr.kuczynski@gmail.com>"からの正しい署名 [不明の]
gpg: *警告*: この鍵は信用できる署名で証明されていません!
gpg: この署名が所有者のものかどうかの検証手段がありません。
主鍵フィンガープリント: 7D2B AF1C F37B 13E2 069D 6956 105B D0E7 3949 9BDB
GPG verified '/usr/local/rvm/archives/rvm-1.29.8.tgz'
Creating group 'rvm'
Installing RVM to /usr/local/rvm/
Installation of RVM in /usr/local/rvm/ is almost complete:

* First you need to add all users that will be using rvm to 'rvm' group,
  and logout - login again, anyone using rvm will be operating with `umask
u=rwx, g=rwx, o=rx`.

* To start using RVM you need to run `source /etc/profile.d/rvm.sh`
  in all your open shell windows, in rare cases you need to reopen all shell
  windows.

* Please do NOT forget to add your users to the rvm group.
  The installer no longer auto-adds root or users to the rvm group. Admins must
  do this.
  Also, please note that group memberships are ONLY evaluated at login time.
  This means that users must log out then back in before group membership takes
  effect!
Thanks for installing RVM
Please consider donating to our open collective to help us maintain RVM.

Donate: https://opencollective.com/rvm/donate

root@raspberrypi:~#
```

4. pi ユーザで動作確認

```
root@raspberrypi:~# exit
ログアウト
pi@raspberrypi:~ $ source /etc/profile.d/rvm.sh
pi@raspberrypi:~ $ rvm -v
rvm 1.29.8 (latest) by Michal Papis, Piotr Kuczynski, Wayne E. Seguin
[https://rvm.io]
pi@raspberrypi:~ $
```

5. rvm の読込と rvm を最新バージョンに更新

```

pi@raspberrypi:~ $ sudo su -

SSH is enabled and the default password for the 'pi' user has not been changed.
This is a security risk - please login as the 'pi' user and type 'passwd' to set a
new password.

root@raspberrypi:~# source /etc/profile.d/rvm.sh
root@raspberrypi:~# rvm get head
Downloading https://get.rvm.io
Downloading https://raw.githubusercontent.com/rvm/rvm/master/binscripts/rvm-
installer.asc
Verifying /usr/local/rvm/archives/rvm-installer.asc
gpg: 2019年07月02日 02時24分59秒 JST に施された署名
gpg: RSA 鍵 62C9E5F4DA300D94AC36166BE206C29FBF04FF17 を使用
gpg: "Michal Papis (RVM signing) <mpapis@gmail.com>" からの正しい署名 [不明の]
gpg: 別名 "Michal Papis <michal.papis@toptal.com>" [不明の]
gpg: 別名 "[jpeg image of size 5015]" [不明の]
gpg: 注意: この鍵は期限切れです!
主鍵フィンガープリント: 409B 6B17 96C2 7546 2A17 0311 3804 BB82 D39D C0E3
副鍵フィンガープリント: 62C9 E5F4 DA30 0D94 AC36 166B E206 C29F BF04 FF17
GPG verified '/usr/local/rvm/archives/rvm-installer'
Downloading https://github.com/rvm/rvm/archive/master.tar.gz
Upgrading the RVM installation in /usr/local/rvm/
Upgrade of RVM in /usr/local/rvm/ is complete.
* Please do NOT forget to add your users to the rvm group.
The installer no longer auto-adds root or users to the rvm group. Admins must
do this.
Also, please note that group memberships are ONLY evaluated at login time.
This means that users must log out then back in before group membership takes
effect!

Thanks for installing RVM
Please consider donating to our open collective to help us maintain RVM.

Donate: https://opencollective.com/rvm/donate

RVM reloaded!
root@raspberrypi:~#

```

6. OS を再起動

```

root@raspberrypi:~# exit
ログアウト
pi@raspberrypi:~ $ sudo reboot

```

7. rvm を sudo を付けて (マルチユーザーとして) インストールする

rvm バージョン確認

```
pi@raspberrypi:~ $ rvm sudo rvm -v
Warning: can not check `/etc/sudoers` for `secure_path`, falling back to call via
`/usr/bin/env`, this breaks rules from `/etc/sudoers`. Run:

export rvm sudo_secure_path=1

to avoid the warning, put it in shell initialization file to make it persistent.

In case there is no `secure_path` in `/etc/sudoers`. Run:

export rvm sudo_secure_path=0

to avoid the warning, put it in shell initialization file to make it persistent.
rvm 1.29.8-next (master) by Michal Papis, Piotr Kuczynski, Wayne E. Seguin
[https://rvm.io]
pi@raspberrypi:~ $
```

インストール可能な ruby のバージョン等を確認する 表示しない場合は再起動

```
pi@raspberrypi:~ $ rvm sudo rvm -v
Warning: can not check `/etc/sudoers` for `secure_path`, falling back to call via
`/usr/bin/env`, this breaks rules from `/etc/sudoers`. Run:

export rvm sudo_secure_path=1

to avoid the warning, put it in shell initialization file to make it persistent.

In case there is no `secure_path` in `/etc/sudoers`. Run:

export rvm sudo_secure_path=0

to avoid the warning, put it in shell initialization file to make it persistent.
rvm 1.29.8-next (master) by Michal Papis, Piotr Kuczynski, Wayne E. Seguin
[https://rvm.io]
pi@raspberrypi:~ $ rvm list known
Warning, new version of rvm available '1.29.8', you are using older version
'1.29.8-next'.
You can disable this warning with: echo rvm_autoupdate_flag=0 >> ~/.rvmrc
You can enable auto-update with: echo rvm_autoupdate_flag=2 >> ~/.rvmrc
You can update manually with: rvm get VERSION (e. g.
'rvm get stable')

# MRI Rubies
[ruby-]1.8.6[-p420]
[ruby-]1.8.7[-head] # security released on head
[ruby-]1.9.1[-p431]
[ruby-]1.9.2[-p330]
[ruby-]1.9.3[-p551]
[ruby-]2.0.0[-p648]
[ruby-]2.1[. 10]
[ruby-]2.2[. 10]
[ruby-]2.3[. 8]
```

```
[ruby-]2.4[.6]
[ruby-]2.5[.5]
[ruby-]2.6[.3]
[ruby-]2.7[.0-preview1]
ruby-head

# for forks use: rvm install ruby-head-<name> --url
https://github.com/github/ruby.git --branch 2.2

# JRuby
jruby-1.6[.8]
jruby-1.7[.27]
jruby-9.1[.17.0]
jruby[-9.2.7.0]
jruby-head

# Rubinius
rbx-1[.4.3]
rbx-2.3[.0]
rbx-2.4[.1]
rbx-2[.5.8]
rbx-3[.107]
rbx-4[.2]
rbx-head

# TruffleRuby
truffleruby[-19.1.0]

# Opal
opal

# Minimalistic ruby implementation - ISO 30170:2012
mruby-1.0.0
mruby-1.1.0
mruby-1.2.0
mruby-1.3.0
mruby-1[.4.1]
mruby-2[.0.1]
mruby[-head]

# Ruby Enterprise Edition
ree-1.8.6
ree[-1.8.7][-2012.02]

# Topaz
topaz

# MagLev
maglev-1.0.0
maglev-1.1[RC1]
maglev[-1.2Alpha4]
maglev-head

# Mac OS X Snow Leopard Or Newer
macruby-0.10
```

```
macruby-0.11
macruby[-0.12]
macruby-nightly
macruby-head

# IronRuby
ironruby[-1.1.3]
ironruby-head
pi@raspberrypi:~ $
```

8. インストールに当たっての必要要件を確認する。

```
pi@raspberrypi:~ $ rvmsudo rvm requirements
Warning: can not check `/etc/sudoers` for `secure_path`, falling back to call via
`/usr/bin/env`, this breaks rules from `/etc/sudoers`. Run:

export rvmsudo_secure_path=1

to avoid the warning, put it in shell initialization file to make it persistent.

In case there is no `secure_path` in `/etc/sudoers`. Run:

export rvmsudo_secure_path=0

to avoid the warning, put it in shell initialization file to make it persistent.
Checking requirements for debian.
Installing requirements for debian.
Updating system.
Installing required packages: gawk, libffi-dev, libgdbm-dev, libncurses5-dev,
libsqlite3-dev, libyaml-dev, sqlite3, libgmp-dev, libssl-dev.....
Requirements installation successful.
pi@raspberrypi:~ $
```

9. 続いて該当バージョン (2.6.3) をインストール

```
pi@raspberrypi:~ $ rvmsudo rvm install 2.6.3
Warning: can not check `/etc/sudoers` for `secure_path`, falling back to call via
`/usr/bin/env`, this breaks rules from `/etc/sudoers`. Run:

export rvmsudo_secure_path=1

to avoid the warning, put it in shell initialization file to make it persistent.

In case there is no `secure_path` in `/etc/sudoers`. Run:

export rvmsudo_secure_path=0

to avoid the warning, put it in shell initialization file to make it persistent.
Warning, new version of rvm available '1.29.8', you are using older version
'1.29.8-next'.
You can disable this warning with: echo rvm_autoupdate_flag=0 >> ~/.rvmrc
You can enable auto-update with: echo rvm_autoupdate_flag=2 >> ~/.rvmrc
You can update manually with: rvm get VERSION (e.g.
'rvm get stable')

Searching for binary rubies, this might take some time.
```

```

No binary rubies available for: debian/10/armhf/ruby-2.6.3.
Continuing with compilation. Please read 'rvm help mount' to get more information
on binary rubies.
Checking requirements for debian.
Requirements installation successful.
Installing Ruby from source to: /usr/local/rvm/rubies/ruby-2.6.3, this may take a
while depending on your cpu(s)...
ruby-2.6.3 - #downloading ruby-2.6.3, this may take a while depending on your
connection...
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left   Speed
100 13.8M  100 13.8M    0     0 4457k      0  0:00:03  0:00:03 --:--:-- 4457k
ruby-2.6.3 - #extracting ruby-2.6.3 to /usr/local/rvm/src/ruby-2.6.3....
ruby-2.6.3 -
#configuring.....
ruby-2.6.3 - #post-configuration..
ruby-2.6.3 -
#compiling.....
.....
ruby-2.6.3 - #installing.....
ruby-2.6.3 - #making binaries executable..
ruby-2.6.3 - #downloading rubygems-3.0.4
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left   Speed
100 862k  100 862k    0     0 2044k      0  --:--:-- --:--:-- --:--:-- 2049k
ruby-2.6.3 - #extracting rubygems-3.0.4....
ruby-2.6.3 - #removing old rubygems.....
ruby-2.6.3 - #installing rubygems-3.0.4.....
ruby-2.6.3 - #gemset created /usr/local/rvm/gems/ruby-2.6.3@global
ruby-2.6.3 - #importing gemset
/usr/local/rvm/gemsets/global.gems.....
.....
ruby-2.6.3 - #generating global wrappers.....
ruby-2.6.3 - #gemset created /usr/local/rvm/gems/ruby-2.6.3
ruby-2.6.3 - #importing gemsetfile /usr/local/rvm/gemsets/default.gems evaluated to
empty gem list
ruby-2.6.3 - #generating default wrappers.....
ruby-2.6.3 - #adjusting #shebangs for (gem irb erb ri rdoc testrb rake).
Install of ruby-2.6.3 - #complete
Ruby was built without documentation, to build it run: rvm docs generate-ri
pi@raspberrypi:~ $

```

10. 最後にパスを通して、また通常使用するバージョンを指定する

```

pi@raspberrypi:~ $ echo '[[ -s "/usr/local/rvm/scripts/rvm" ]] && .
"/usr/local/rvm/scripts/rvm" # Load RVM function' >> ~/.bash_profile
pi@raspberrypi:~ $ rvm use 2.6.3 --default
Using /usr/local/rvm/gems/ruby-2.6.3
pi@raspberrypi:~ $

```

11. システム起動時の ruby デフォルトを 2.6.3 とする

```

pi@raspberrypi:~ $ rvmsudo rvm alias create default ruby-2.6.3
Warning: can not check `/etc/sudoers` for `secure_path`, falling back to call via
`/usr/bin/env`, this breaks rules from `/etc/sudoers`. Run:

```

```
export rvmsudo_secure_path=1
```

to avoid the warning, put it in shell initialization file to make it persistent.

In case there is no `secure_path` in `/etc/sudoers`. Run:

```
export rvmsudo_secure_path=0
```

to avoid the warning, put it in shell initialization file to make it persistent.

Creating alias default for ruby-2.6.3....

```
pi@raspberrypi:~ $
```

1 2. install されている ruby の確認

```
pi@raspberrypi:~ $ rvm list
```

```
=* ruby-2.6.3 [ armv7l ]
```

```
# => - current
```

```
# =* - current && default
```

```
# * - default
```

```
pi@raspberrypi:~ $
```

13. 関連モジュールのインストール手順

libpq-dev が必要なのでインストール

```

pi@raspberrypi:~ $ sudo apt-get install libpq-dev
パッケージリストを読み込んでいます... 完了
依存関係ツリーを作成しています
状態情報を読み取っています... 完了
提案パッケージ:
  postgresql-doc-11
以下のパッケージが新たにインストールされます:
  libpq-dev
アップグレード: 0 個、新規インストール: 1 個、削除: 0 個、保留: 0 個。
148 kB のアーカイブを取得する必要があります。
この操作後に追加で 567 kB のディスク容量が消費されます。
取得:1 http://ftp.jaist.ac.jp/pub/Linux/raspbian-archive/raspbian buster/main armhf
libpq-dev armhf 11.4-1 [148 kB]
148 kB を 2 秒 で取得しました (84.0 kB/s)
以前に未選択のパッケージ libpq-dev を選択しています。
(データベースを読み込んでいます ... 現在 46294 個のファイルとディレクトリがインストールされています。)
.../libpq-dev_11.4-1_armhf.deb を展開する準備をしています ...
libpq-dev (11.4-1) を展開しています...
libpq-dev (11.4-1) を設定しています ...
man-db (2.8.5-2) のトリガを処理しています ...
pi@raspberrypi:~ $

```

pg のインストール

```

pi@raspberrypi:~ $ rvmsudo gem install pg --no-document
Warning: can not check `/etc/sudoers` for `secure_path`, falling back to call via
`/usr/bin/env`, this breaks rules from `/etc/sudoers`. Run:

  export rvmsudo_secure_path=1

to avoid the warning, put it in shell initialization file to make it persistent.

In case there is no `secure_path` in `/etc/sudoers`. Run:

  export rvmsudo_secure_path=0

to avoid the warning, put it in shell initialization file to make it persistent.
Fetching pg-1.1.4.gem
Building native extensions. This could take a while...
Successfully installed pg-1.1.4
1 gem installed
pi@raspberrypi:~ $

```

dbi のインストール

```

pi@raspberrypi:~ $ rvmsudo gem install dbi --no-document
Warning: can not check `/etc/sudoers` for `secure_path`, falling back to call via
`/usr/bin/env`, this breaks rules from `/etc/sudoers`. Run:

  export rvmsudo_secure_path=1

to avoid the warning, put it in shell initialization file to make it persistent.

```

In case there is no `secure_path` in `/etc/sudoers`. Run:

```
export rvsudo_secure_path=0
```

to avoid the warning, put it in shell initialization file to make it persistent.

```
Fetching dbi-0.4.5.gem
```

```
Fetching deprecated-2.0.1.gem
```

```
Successfully installed deprecated-2.0.1
```

```
Successfully installed dbi-0.4.5
```

```
2 gems installed
```

```
pi@raspberrypi:~ $
```

sequel のインストール

```
pi@raspberrypi:~ $ rvsudo gem install sequel --no-document
```

```
Warning: can not check `/etc/sudoers` for `secure_path`, falling back to call via  
`/usr/bin/env`, this breaks rules from `/etc/sudoers`. Run:
```

```
export rvsudo_secure_path=1
```

to avoid the warning, put it in shell initialization file to make it persistent.

In case there is no `secure_path` in `/etc/sudoers`. Run:

```
export rvsudo_secure_path=0
```

to avoid the warning, put it in shell initialization file to make it persistent.

```
Fetching sequel-5.22.0.gem
```

```
Successfully installed sequel-5.22.0
```

```
1 gem installed
```

```
pi@raspberrypi:~ $
```

rack のインストール

```
pi@raspberrypi:~ $ rvsudo gem install rack --no-document
```

```
Warning: can not check `/etc/sudoers` for `secure_path`, falling back to call via  
`/usr/bin/env`, this breaks rules from `/etc/sudoers`. Run:
```

```
export rvsudo_secure_path=1
```

to avoid the warning, put it in shell initialization file to make it persistent.

In case there is no `secure_path` in `/etc/sudoers`. Run:

```
export rvsudo_secure_path=0
```

to avoid the warning, put it in shell initialization file to make it persistent.

```
Fetching rack-2.0.7.gem
```

```
Successfully installed rack-2.0.7
```

```
1 gem installed
```

```
pi@raspberrypi:~ $
```

prawn のインストール

PDF を使うなら `rvmsudo gem install prawn --version "=0.13.1" --no-document`

```
pi@raspberrypi:~ $ rvmsudo gem install prawn --no-document
Warning: can not check `/etc/sudoers` for `secure_path`, falling back to call via
`/usr/bin/env`, this breaks rules from `/etc/sudoers`. Run:

    export rvmsudo_secure_path=1

to avoid the warning, put it in shell initialization file to make it persistent.

In case there is no `secure_path` in `/etc/sudoers`. Run:

    export rvmsudo_secure_path=0

to avoid the warning, put it in shell initialization file to make it persistent.
Fetching prawn-2.2.2.gem
Fetching pdf-core-0.7.0.gem
Fetching ttfunk-1.5.1.gem
Successfully installed pdf-core-0.7.0
Successfully installed ttfunk-1.5.1
Successfully installed prawn-2.2.2
3 gems installed
pi@raspberrypi:~ $
```

prawn-svg のインストール

```
pi@raspberrypi:~ $ rvmsudo gem install prawn-svg --no-document
Warning: can not check `/etc/sudoers` for `secure_path`, falling back to call via
`/usr/bin/env`, this breaks rules from `/etc/sudoers`. Run:

    export rvmsudo_secure_path=1

to avoid the warning, put it in shell initialization file to make it persistent.

In case there is no `secure_path` in `/etc/sudoers`. Run:

    export rvmsudo_secure_path=0

to avoid the warning, put it in shell initialization file to make it persistent.
Fetching css_parser-1.7.0.gem
Fetching public_suffix-3.1.1.gem
Fetching addressable-2.6.0.gem
Fetching prawn-svg-0.29.1.gem
Successfully installed public_suffix-3.1.1
Successfully installed addressable-2.6.0
Successfully installed css_parser-1.7.0
Successfully installed prawn-svg-0.29.1
4 gems installed
pi@raspberrypi:~ $
```

victor のインストール

```
pi@raspberrypi:~ $ rvmsudo gem install victor --no-document
Warning: can not check `/etc/sudoers` for `secure_path`, falling back to call via
`/usr/bin/env`, this breaks rules from `/etc/sudoers`. Run:
```

```
export rvmsudo_secure_path=1
```

to avoid the warning, put it in shell initialization file to make it persistent.

In case there is no `secure_path` in `/etc/sudoers`. Run:

```
export rvmsudo_secure_path=0
```

to avoid the warning, put it in shell initialization file to make it persistent.

```
Fetching victor-0.2.7.gem
Successfully installed victor-0.2.7
1 gem installed
pi@raspberrypi:~ $
```

rubyXL のインストール

```
pi@raspberrypi:~ $ rvmsudo gem install rubyXL --no-document
Warning: can not check `/etc/sudoers` for `secure_path`, falling back to call via
`/usr/bin/env`, this breaks rules from `/etc/sudoers`. Run:
```

```
export rvmsudo_secure_path=1
```

to avoid the warning, put it in shell initialization file to make it persistent.

In case there is no `secure_path` in `/etc/sudoers`. Run:

```
export rvmsudo_secure_path=0
```

to avoid the warning, put it in shell initialization file to make it persistent.

```
Fetching rubyzip-1.2.3.gem
Fetching nokogiri-1.10.3.gem
Fetching rubyXL-3.4.5.gem
Fetching mini_portile2-2.4.0.gem
Successfully installed mini_portile2-2.4.0
Building native extensions. This could take a while...
Successfully installed nokogiri-1.10.3
Successfully installed rubyzip-1.2.3
Successfully installed rubyXL-3.4.5
4 gems installed
pi@raspberrypi:~ $
```

rqrqcode のインストール

```
pi@raspberrypi:~ $ rvmsudo gem install rqrqcode --no-document
Warning: can not check `/etc/sudoers` for `secure_path`, falling back to call via
`/usr/bin/env`, this breaks rules from `/etc/sudoers`. Run:
```

```
export rvmsudo_secure_path=1
```

to avoid the warning, put it in shell initialization file to make it persistent.

In case there is no `secure_path` in `/etc/sudoers`. Run:

```
export rvmsudo_secure_path=0
```

to avoid the warning, put it in shell initialization file to make it persistent.

```
Fetching chunky_png-1.3.11.gem
```

```
Fetching rqrqcode-0.10.1.gem
```

```
Successfully installed chunky_png-1.3.11
```

```
Successfully installed rqrqcode-0.10.1
```

```
2 gems installed
```

```
pi@raspberrypi:~ $
```

第 10 章 Passenger のインストール

1. passenger のインストール手順

passenger の gem インストール

```
pi@raspberrypi:~ $ rvmsudo gem install passenger --no-document
Warning: can not check `/etc/sudoers` for `secure_path`, falling back to call via
`/usr/bin/env`, this breaks rules from `/etc/sudoers`. Run:

    export rvmsudo_secure_path=1

to avoid the warning, put it in shell initialization file to make it persistent.

In case there is no `secure_path` in `/etc/sudoers`. Run:

    export rvmsudo_secure_path=0

to avoid the warning, put it in shell initialization file to make it persistent.
Fetching passenger-6.0.2.gem
Building native extensions. This could take a while...
Successfully installed passenger-6.0.2
1 gem installed
pi@raspberrypi:~ $
```

libcurl4-openssl-dev のインストール

```
pi@raspberrypi:~ $ sudo apt-get install libcurl4-openssl-dev
パッケージリストを読み込んでいます... 完了
依存関係ツリーを作成しています
状態情報を読み取っています... 完了
提案パッケージ:
  libcurl4-doc libidn11-dev libkrb5-dev librtmp-dev libssh2-1-dev
以下のパッケージが新たにインストールされます:
  libcurl4-openssl-dev
アップグレード: 0 個、新規インストール: 1 個、削除: 0 個、保留: 0 個。
366 kB のアーカイブを取得する必要があります。
この操作後に追加で 1,105 kB のディスク容量が消費されます。
取得:1 http://ftp.jaist.ac.jp/pub/Linux/raspbian-archive/raspbian buster/main armhf
libcurl4-openssl-dev armhf 7.64.0-4 [366 kB]
366 kB を 1 秒 で取得しました (253 kB/s)
以前に未選択のパッケージ libcurl4-openssl-dev:armhf を選択しています。
(データベースを読み込んでいます ... 現在 46320 個のファイルとディレクトリがインスト
ールされています。)
.../libcurl4-openssl-dev_7.64.0-4_armhf.deb を展開する準備をしています ...
libcurl4-openssl-dev:armhf (7.64.0-4) を展開しています...
libcurl4-openssl-dev:armhf (7.64.0-4) を設定しています ...
man-db (2.8.5-2) のトリガを処理しています ...
pi@raspberrypi:~ $
```

passenger install

```
pi@raspberrypi:~$ rvmsudo passenger-install-apache2-module --auto --languages ruby
Warning: can not check `/etc/sudoers` for `secure_path`, falling back to call via
`/usr/bin/env`, this breaks rules from `/etc/sudoers`. Run:
```

```
export rvmsudo_secure_path=1
```

to avoid the warning, put it in shell initialization file to make it persistent.

In case there is no `secure_path` in `/etc/sudoers`. Run:

```
export rvmsudo_secure_path=0
```

to avoid the warning, put it in shell initialization file to make it persistent.

Welcome to the Phusion Passenger Apache 2 module installer, v6.0.2.

This installer will guide you through the entire installation process. It shouldn't take more than 3 minutes in total.

Here's what you can expect from the installation process:

1. The Apache 2 module will be installed for you.
2. You'll learn how to configure Apache.
3. You'll learn how to deploy a Ruby on Rails application.

Don't worry if anything goes wrong. This installer will advise you on how to solve any problems.

Press Enter to continue, or Ctrl-C to abort.

途中省略

```
checking for alloca.h... yes
checking for ruby/version.h... yes
checking for ruby/io.h... yes
checking for ruby/thread.h... yes
checking for ruby_version... no
checking for rb_thread_io_blocking_region() in ruby/io.h... yes
checking for rb_thread_call_without_gvl() in ruby/thread.h... yes
creating Makefile
cd `buildout/ruby/ruby-2.6.3-armv7l-linux/` && make
compiling /usr/local/rvm/gems/ruby-2.6.3/gems/passenger-
6.0.2/src/ruby_native_extension/passenger_native_support.c
linking shared-object passenger_native_support.so
```

Almost there!

Please edit your Apache configuration file, and add these lines:

```
LoadModule passenger_module /usr/local/rvm/gems/ruby-2.6.3/gems/passenger-
6.0.2/buildout/apache2/mod_passenger.so
<IfModule mod_passenger.c>
    PassengerRoot /usr/local/rvm/gems/ruby-2.6.3/gems/passenger-6.0.2
    PassengerDefaultRuby /usr/local/rvm/gems/ruby-2.6.3/wrappers/ruby
```

```
</IfModule>
```

After you restart Apache, you are ready to deploy any number of web applications on Apache, with a minimum amount of configuration!

Press ENTER when you are done editing.

```
-----  
Validating installation...
```

- * Checking whether this Passenger install is in PATH... ✓
- * Checking whether there are no other Passenger installations... ✓
- * Checking whether Apache is installed... ✓
- * Checking whether the Passenger module is correctly configured in Apache... (!)

You did not specify 'LoadModule passenger_module' in any of your Apache configuration files. Please paste the configuration snippet that this installer printed earlier, into one of your Apache configuration files, such as /etc/apache2/apache2.conf.

Detected 0 error(s), 1 warning(s).
Press ENTER to continue.

```
-----  
Deploying a web application
```

To learn how to deploy a web app on Passenger, please follow the deployment guide:

<https://www.phusionpassenger.com/library/deploy/apache/deploy/>

Enjoy Phusion Passenger, a product of Phusion® (www.phusion.nl) :-)
<https://www.phusionpassenger.com>

Passenger® is a registered trademark of Phusion Holding B.V.
pi@raspberrypi:~ \$

ビルドしたモジュールのパスをデフォルト値と差し替えます

```
pi@raspberrypi:~$ sudo vi /etc/apache2/mods-enabled/passenger.load  
LoadModule passenger_module /usr/local/rvm/gems/ruby-2.6.3/gems/passenger-  
6.0.2/buildout/apache2/mod_passenger.so
```

apache の設定

```
pi@raspberrypi:~$ sudo vi /etc/apache2/sites-enabled/rmenu.conf  
<IfModule mod_passenger.c>  
  PassengerRoot /usr/local/rvm/gems/ruby-2.6.3/gems/passenger-6.0.2  
  PassengerDefaultRuby /usr/local/rvm/gems/ruby-2.6.3/wrappers/ruby  
</IfModule>  
  
PassengerMaxPoolSize 20
```

```

PassengerMaxInstancesPerApp 4
PassengerPoolIdleTime 3600
PassengerHighPerformance on
PassengerStatThrottleRate 10
RailsSpawnMethod smart
RailsAppSpawnerIdleTime 86400
PassengerMaxPreloaderIdleTime 0

RackBaseURI /rmenu_user

```

パッセンジャー追加設定、一時ファイル等、

```

pi@raspberrypi:~$ sudo su - rmenu_user
rmenu_user@raspberrypi:~$ cd /home/rmenu_user/Rmenu
rmenu_user@raspberrypi:~/Rmenu$ mkdir public logs tmp
rmenu_user@raspberrypi:~/Rmenu$ touch /home/rmenu_user/Rmenu/tmp/restart.txt
rmenu_user@raspberrypi:~/Rmenu$ exit
ログアウト

```

ドキュメントルート配下にシンボリックリンクを置く

```

pi@raspberrypi:~$ sudo ln -s /home/rmenu_user/Rmenu/public /var/www/html/rmenu_user
pi@raspberrypi:~$

```

2. 設定の確認

設定の確認

```

pi@raspberrypi:~$ sudo apachectl configtest
AH00558: apache2: Could not reliably determine the server's fully qualified domain
name, using 127.0.1.1. Set the 'ServerName' directive globally to suppress this
message
Syntax OK
pi@raspberrypi:~$

```

3. passenger を起動

apache の設定 再読み込みと再起動。

```

pi@raspberrypi:~$ sudo /etc/init.d/apache2 reload
[ ok ] Reloading apache2 configuration (via systemctl): apache2.service.
pi@raspberrypi:~$

```

4. SD カード保護のため swap メモリを 0 にする

swap 削除前ディスク領域確認 残り 614M 使用 92%

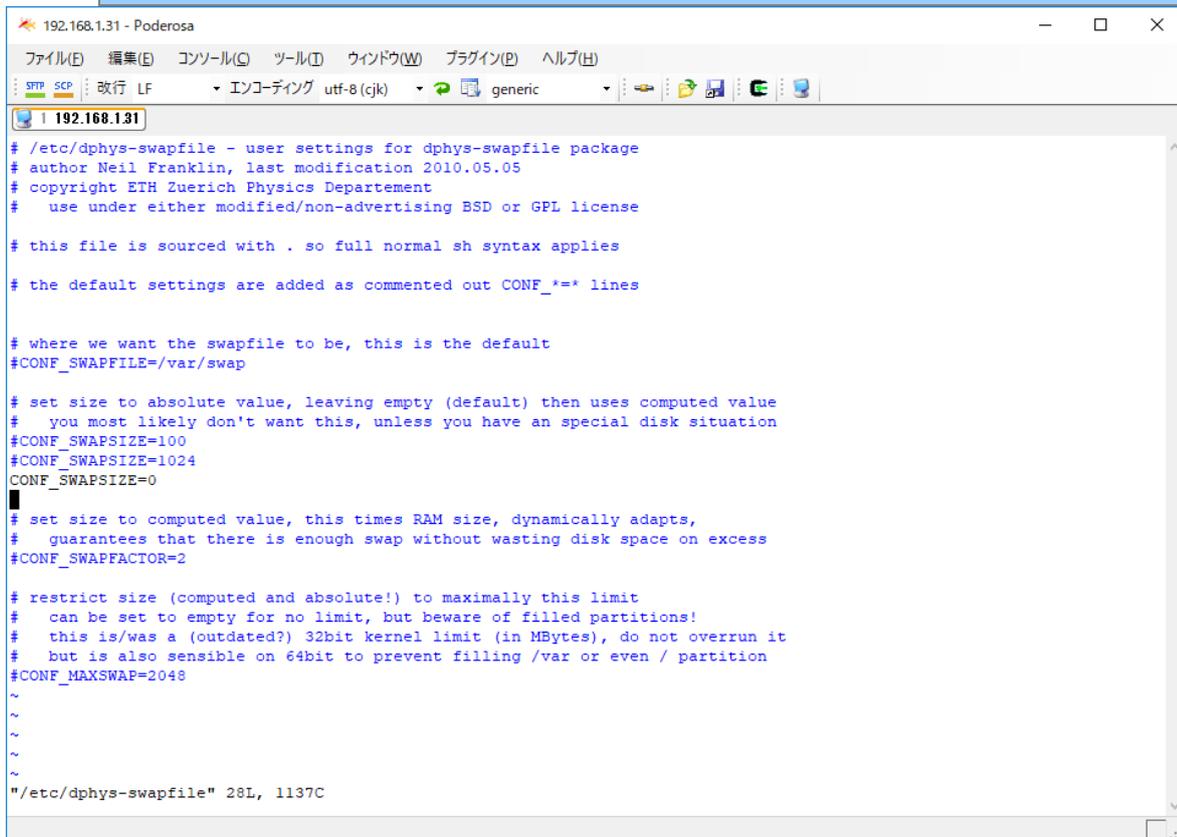
```

pi@raspberrypi:~$ df -h
ファイルシステム サイズ 使用 残り 使用% マウント位置
/dev/root          7.3G  6.4G  614M   92% /
devtmpfs           460M    0  460M    0% /dev
tmpfs              464M  4.0K  464M    1% /dev/shm
tmpfs              464M   13M  452M    3% /run
tmpfs              5.0M  4.0K  5.0M    1% /run/lock
tmpfs              464M    0  464M    0% /sys/fs/cgroup
/dev/mmcblk0p1     44M   22M   22M   50% /boot
tmpfs              93M    0   93M    0% /run/user/1001
tmpfs              93M    0   93M    0% /run/user/1000

```

```
$ sudo vi /etc/dphys-swapfile
```

```
#CONF_SWAPSIZE=100  
#CONF_SWAPSIZE=1024  
CONF_SWAPSIZE=0
```



```
# /etc/dphys-swapfile - user settings for dphys-swapfile package  
# author Neil Franklin, last modification 2010.05.05  
# copyright ETH Zuerich Physics Departement  
# use under either modified/non-advertising BSD or GPL license  
  
# this file is sourced with . so full normal sh syntax applies  
  
# the default settings are added as commented out CONF_*=* lines  
  
# where we want the swapfile to be, this is the default  
#CONF_SWAPFILE=/var/swap  
  
# set size to absolute value, leaving empty (default) then uses computed value  
# you most likely don't want this, unless you have an special disk situation  
#CONF_SWAPSIZE=100  
#CONF_SWAPSIZE=1024  
CONF_SWAPSIZE=0  
█  
# set size to computed value, this times RAM size, dynamically adapts,  
# guarantees that there is enough swap without wasting disk space on excess  
#CONF_SWAPFACTOR=2  
  
# restrict size (computed and absolute!) to maximally this limit  
# can be set to empty for no limit, but beware of filled partitions!  
# this is/was a (outdated?) 32bit kernel limit (in MBytes), do not overrun it  
# but is also sensible on 64bit to prevent filling /var or even / partition  
#CONF_MAXSWAP=2048  
~  
~  
~  
~  
~  
"/etc/dphys-swapfile" 28L, 1137C
```

```
pi@raspberrypi:~ $ sudo vi /etc/dphys-swapfile
```

```
pi@raspberrypi:~ $ sudo dphys-swapfile install
```

```
want /var/swap=0MByte, checking existing: deleting wrong size file (1073741824),
```

```
generating swapfile ... pi@raspberrypi:~ $ sudo dphys-swapfile swapon
```

```
/sbin/dphys-swapfile: ERROR: swap file /var/swap missing! you need to first run  
/sbin/dphys-swapfile setup to generate one
```

```
pi@raspberrypi:~ $ free
```

	total	used	free	shared	buff/cache	available
Mem:	949444	100756	505916	23388	342772	773864
Swap:	0	0	0			

```
pi@raspberrypi:~ $
```

第 1 1 章 サーバ電源投入時・Rmenu 起動スクリプトの自動化

1. 起動スクリプト

/etc/rc.d/rc.local の exit0 直前に次の 1 行を追記

```
su - rmenu_user -c "cd Rmenu/System ; ./restart_server.tcsh"
```

```
pi@raspberrypi:~$ sudo vi /etc/rc.local
#!/bin/sh -e
#
# rc.local
#
# This script is executed at the end of each multiuser runlevel.
# Make sure that the script will "exit 0" on success or any other
# value on error.
#
# In order to enable or disable this script just change the execution
# bits.
#
# By default this script does nothing.

# Print the IP address
_IP=$(hostname -I) || true
if [ "$_IP" ]; then
    printf "My IP address is %s\n" "$_IP"
fi
su - rmenu_user -c "cd Rmenu/System ; ./restart_server.tcsh"
exit 0
```

2. 起動設定

RaspberryPI では rc.local は初期状態では使えない

参考 URL: 【Raspberry Pi】 /etc/rc.local が実行されない問題について
<http://peacejet.hatenablog.com/entry/2017/07/08/000907>

3. コンフィグ画面起動

```
pi@raspberrypi:~$ sudo raspi-config
```

4. 「Boot Options」を選択します。
5. 「B2 Wait for Network at boot」を選択します。
6. <はい>を選択します。
7. <了解>を選択します。
8. リブートしてください。

```
pi@raspberrypi:~$ sudo reboot
```

第 1 2 章 Rmenu Application のインストールと起動

1. rmenu_user フォルダの所有グループ・パーミションを変更

[ユーザ root で行います]

/home/rmenu_user フォルダの所有グループを変更します user : rmenu_user

```
pi@raspberrypi:~ $ cd /home/rmenu_user/
pi@raspberrypi:/home/rmenu_user$ ls -al
合計 24
drwxr-xr-x 3 rmenu_user www-data 4096 10月 2 20:42 .
drwxr-xr-x 5 root root 4096 10月 2 17:16 ..
-rw-r--r-- 1 rmenu_user www-data 220 10月 2 17:16 .bash_logout
-rw-r--r-- 1 rmenu_user www-data 3523 10月 2 17:16 .bashrc
-rw-r--r-- 1 rmenu_user www-data 675 10月 2 17:16 .profile
drwxr-xr-x 7 rmenu_user rmenu_user 4096 10月 2 20:42 Rmenu
pi@raspberrypi:/home/rmenu_user$ sudo chown -R rmenu_user:www-data /home/rmenu_user
pi@raspberrypi:/home/rmenu_user$ ls -al
合計 24
drwxr-xr-x 3 rmenu_user www-data 4096 10月 2 20:42 .
drwxr-xr-x 5 root root 4096 10月 2 17:16 ..
-rw-r--r-- 1 rmenu_user www-data 220 10月 2 17:16 .bash_logout
-rw-r--r-- 1 rmenu_user www-data 3523 10月 2 17:16 .bashrc
-rw-r--r-- 1 rmenu_user www-data 675 10月 2 17:16 .profile
drwxr-xr-x 7 rmenu_user www-data 4096 10月 2 20:42 Rmenu
pi@raspberrypi:/home/rmenu_user$
```

/home/rmenu_user フォルダのパーミションを変更します

```
pi@raspberrypi:/home/rmenu_user$ sudo chmod -R g+wx /home/rmenu_user
pi@raspberrypi:/home/rmenu_user$ ls -al
合計 24
drwxrwxr-x 3 rmenu_user www-data 4096 10月 2 20:42 .
drwxr-xr-x 5 root root 4096 10月 2 17:16 ..
-rw-rwxr-- 1 rmenu_user www-data 220 10月 2 17:16 .bash_logout
-rw-rwxr-- 1 rmenu_user www-data 3523 10月 2 17:16 .bashrc
-rw-rwxr-- 1 rmenu_user www-data 675 10月 2 17:16 .profile
drwxrwxr-x 7 rmenu_user www-data 4096 10月 2 20:42 Rmenu
pi@raspberrypi:/home/rmenu_user$
```

[ユーザ rmenu_user で行います]

/home/rmenu_user/Rmenu/System 実行権限を付与します

```
pi@raspberrypi:/home/rmenu_user$ sudo su - rmenu_user
rmenu_user@raspberrypi:~ $ ls -al
合計 24
drwxrwxr-x 3 rmenu_user www-data 4096 10月 2 20:42 .
drwxr-xr-x 5 root root 4096 10月 2 17:16 ..
-rw-rwxr-- 1 rmenu_user www-data 220 10月 2 17:16 .bash_logout
-rw-rwxr-- 1 rmenu_user www-data 3523 10月 2 17:16 .bashrc
-rw-rwxr-- 1 rmenu_user www-data 675 10月 2 17:16 .profile
drwxrwxr-x 7 rmenu_user www-data 4096 10月 2 20:42 Rmenu
rmenu_user@raspberrypi:~ $ find . \$( -name '*.sh' -o -name '*.tclsh' ) -exec chmod
+x {} \;
rmenu_user@raspberrypi:~ $ ls -al Rmenu/System/
合計 132
drwxrwxr-x 4 rmenu_user www-data 4096 10月 2 20:42 .
```

```

drwxrwxr-x 7 rmenu_user www-data 4096 10月 2 20:42 ..
drwxrwxr-x 5 rmenu_user www-data 4096 10月 2 20:42 Html
-rw-rwxr-- 1 rmenu_user www-data 159 10月 2 20:42 MainJsonEditor.bat
drwxrwxr-x 4 rmenu_user www-data 4096 10月 2 20:43 Server
-rw-rwxr-- 1 rmenu_user www-data 89 10月 2 20:42 no0_Rackup.bat
-rw-rwxr-- 1 rmenu_user www-data 147 10月 2 20:42 no11_TupleSpace.bat
-rwxrwxr-x 1 rmenu_user www-data 1000 10月 2 20:42 no11_TupleSpace.sh
-rw-rwxr-- 1 rmenu_user www-data 144 10月 2 20:42 no12_TupleServer.bat
-rwxrwxr-x 1 rmenu_user www-data 888 10月 2 20:42 no12_TupleServer.sh
-rw-rwxr-- 1 rmenu_user www-data 156 10月 2 20:42 no13_TupleClient.bat
-rwxrwxr-x 1 rmenu_user www-data 984 10月 2 20:42 no13_TupleClient.sh
-rw-rwxr-- 1 rmenu_user www-data 149 10月 2 20:42 no18_TupleServerEnd.bat
-rwxrwxr-x 1 rmenu_user www-data 401 10月 2 20:42 no18_TupleServerEnd.sh
-rw-rwxr-- 1 rmenu_user www-data 161 10月 2 20:42 no19_TupleClientEnd.bat
-rwxrwxr-x 1 rmenu_user www-data 410 10月 2 20:42 no19_TupleClientEnd.sh
-rw-rwxr-- 1 rmenu_user www-data 155 10月 2 20:42 no1_MainController.bat
-rwxrwxr-x 1 rmenu_user www-data 1081 10月 2 20:42 no1_MainController.sh
-rw-rwxr-- 1 rmenu_user www-data 142 10月 2 20:42 no2_MainModel.bat
-rwxrwxr-x 1 rmenu_user www-data 996 10月 2 20:42 no2_MainModel.sh
-rw-rwxr-- 1 rmenu_user www-data 135 10月 2 20:42 no3_MainView.bat
-rwxrwxr-x 1 rmenu_user www-data 995 10月 2 20:42 no3_MainView.sh
-rw-rwxr-- 1 rmenu_user www-data 148 10月 2 20:42 no4_PrintServer.bat
-rwxrwxr-x 1 rmenu_user www-data 1175 10月 2 20:42 no4_PrintServer.sh
-rw-rwxr-- 1 rmenu_user www-data 147 10月 2 20:42 nobat11_TupleSpace .bat
-rwxrwxr-x 1 rmenu_user www-data 1000 10月 2 20:42 nobat11_TupleSpace.sh
-rw-rwxr-- 1 rmenu_user www-data 144 10月 2 20:42 nobat12_TupleServer.bat
-rwxrwxr-x 1 rmenu_user www-data 888 10月 2 20:42 nobat12_TupleServer.sh
-rw-rwxr-- 1 rmenu_user www-data 164 10月 2 20:42 nobat13_TupleParallelServer.bat
-rwxrwxr-x 1 rmenu_user www-data 992 10月 2 20:42 nobat13_TupleParallelServer.sh
-rw-rwxr-- 1 rmenu_user www-data 164 10月 2 20:42 nobat14_TupleParallelClient.bat
-rwxrwxr-x 1 rmenu_user www-data 992 10月 2 20:42 nobat14_TupleParallelClient.sh
-rwxrwxr-x 1 rmenu_user www-data 1551 10月 2 20:42 restart_server.tcsh
rmenu_user@raspberrypi:~ $

```

restart_server.tcsh を起動する

```

rmenu_user@raspberrypi:~ $ cd Rmenu/System/
rmenu_user@raspberrypi:~/Rmenu/System $ ./restart_server.tcsh
カレントディレクトリ変更
/home/rmenu_user/Rmenu/System/Server/Libraries/Main
ダブルクライアントを終了させる
druby://localhost:12349 - #<Errno::ECONNREFUSED: Connection refused - connect(2)
for "localhost" port 12349>
カレントディレクトリ変更
/home/rmenu_user/Rmenu/System/Server/Libraries/Main
ダブルサーバを終了させる
druby://localhost:12349 - #<Errno::ECONNREFUSED: Connection refused - connect(2)
for "localhost" port 12349>
[1] 1620
カレントディレクトリ変更
/home/rmenu_user/Rmenu/System/Server/Libraries/Main
ダブルスペースを起動する
URI:druby://localhost:12349
TapleSpace ready
[2] 1625

```

```
カレントディレクトリ変更
/home/rmenu_user/Rmenu/System/Server/Libraries/Main
タプルサーバを起動する
[3] 1631
カレントディレクトリ変更
/home/rmenu_user/Rmenu/System/Server/Libraries/Main
タプルクライアントを起動する
TapleServer ready
[4] 1641
カレントディレクトリ変更
/home/rmenu_user/Rmenu/System/Server/Libraries/Main
プリントサーバを起動する
TapleClient ready
[5] 1653
rmenu_user@raspberrypi:~/Rmenu/System $ カレントディレクトリ変更
/home/rmenu_user/Rmenu/System/Server/Libraries/Main
メインコントローラを起動する
druby://localhost:12347
PrintServer ready
druby://localhost:12345
MainController ready

rmenu_user@raspberrypi:~/Rmenu/System $
```

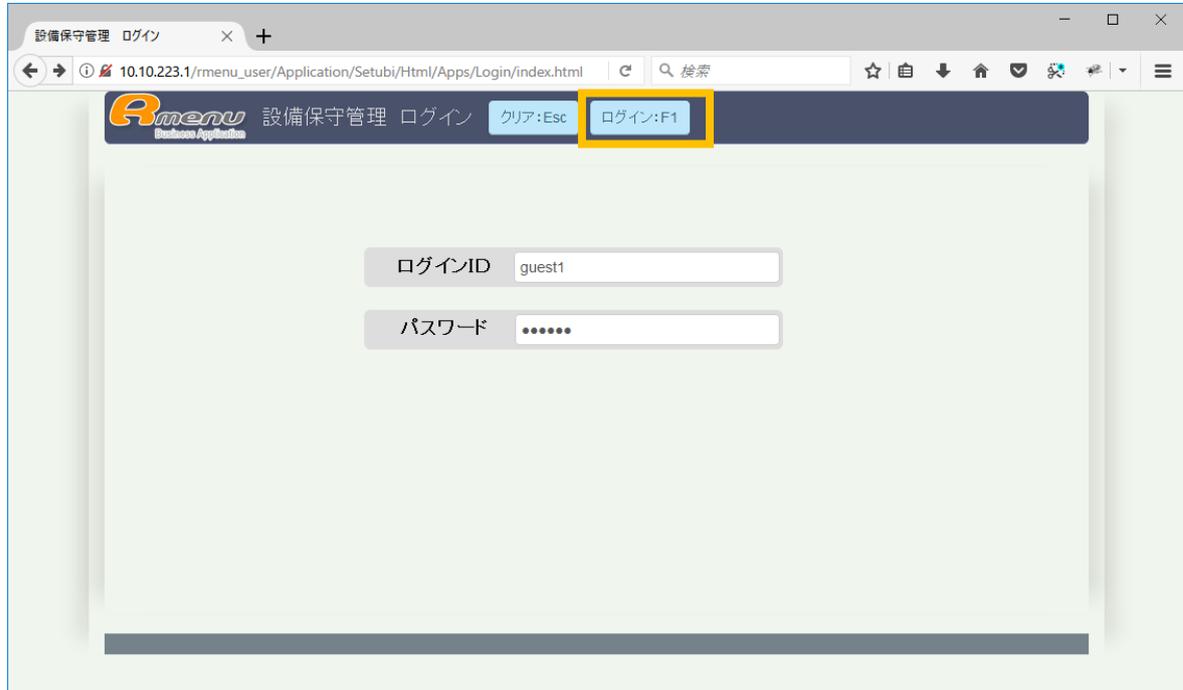
第 1 3 章 Setubi Application 起動

Firefox で次の URL を開く (dhcp で付与されたアドレスに読み換えてください。)
 http://ww.xx.yy.zz/rmenu_user/Application/Setubi/Html/Apps/Login/index.html

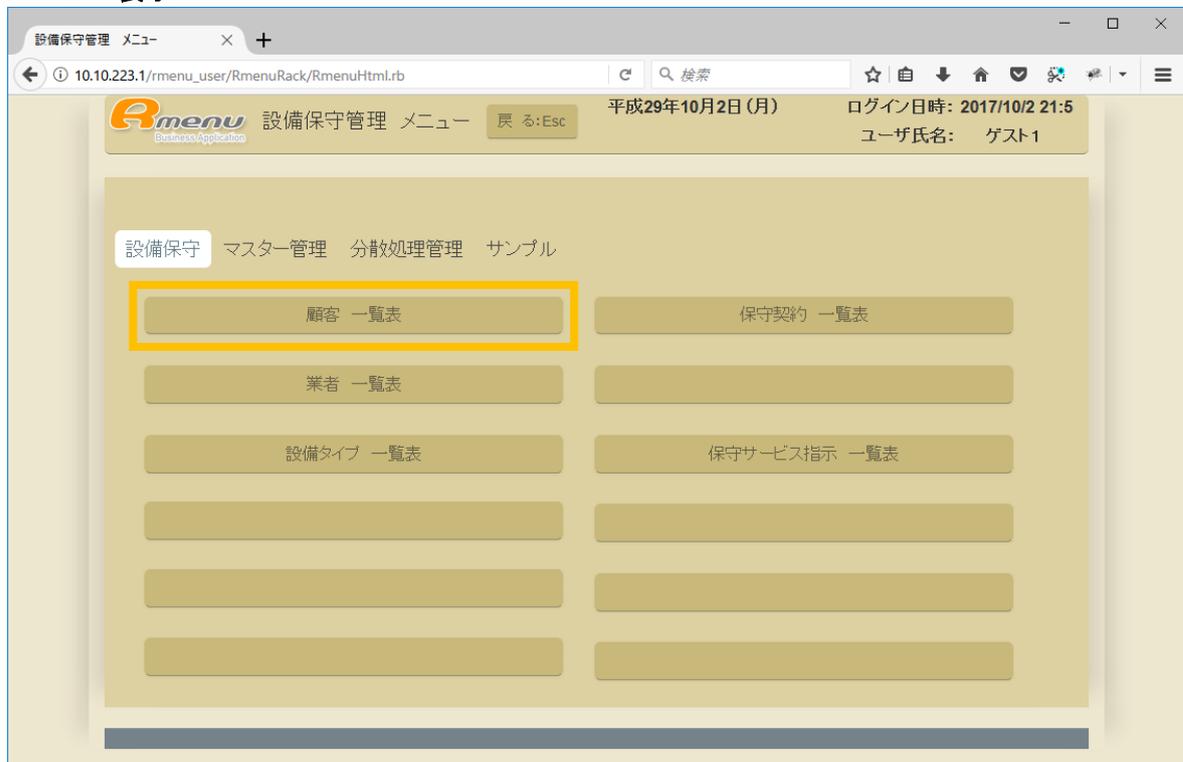
id:guest1、pw:guest1

id:guest2、pw:guest2

ログイン



メニュー表示



「顧客 一覧表」 クリック

第 14 章 注意事項

1. Raspberry PI の性能と互換性

初代の Raspberry PI と現行の Raspberry PI 3 では約 6 倍の性能差があります。メモリの搭載量も 512MByte と 1GByte の違いがあるため CPU 性能、メモリを必要とするコンパイルでは、古い機種において性能不足で途中でフリーズする場合がありますが、SD カードに書き込んだ OS はどの機種でも問題なく動作します。