

# Raspberry Pi で OpenFOAMの環境構築



中山 勝之 (オープンCAE勉強会@富山)

## 内容

Raspberry PiでOpenFOAMの環境構築を行った

一般的なPCに対する環境構築とは若干異なり、そのため試行錯誤した

本資料はインストールメモをまとめたもの

## 構築環境

使用マシン: Raspberry Pi 3 Model B

SoC: Broadcom BCM2837 1.2 GHz ARM Cortex-A53 クアッドコアプロセッサ

メモリ: 1 GB LPDDR2

microSD カード容量 : 32 GB

OS: Ubuntu MATE 16.04 LTS (32bit)

OpenFOAM Ver.: 3.0.1

01

# Raspberry PiにOSをインストール

# Raspberry Piにインストール可能なOS

<https://www.raspberrypi.org/downloads/>

## DOWNLOADS

**Raspbian** is the Foundation's official supported Operating System. Download it here, or use **NOOBS**, our easy installer for Raspbian and more.



NOOBS



RASPBIAN

## THIRD PARTY OPERATING SYSTEM IMAGES

Third party images are also available:



UBUNTU MATE



SNAPPY UBUNTU CORE



WINDOWS 10 IOT CORE



OSMC



LIBREELEC



PINET



RISC OS



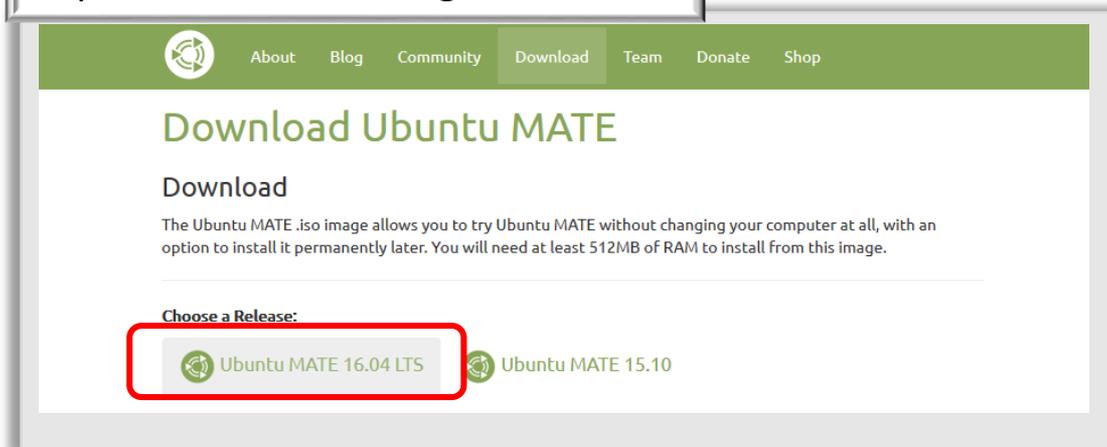
WEATHER STATION

今回はOpenFOAMでスタンダードで  
使用されるUbuntu系OSである  
Ubuntu MATEをインストールする

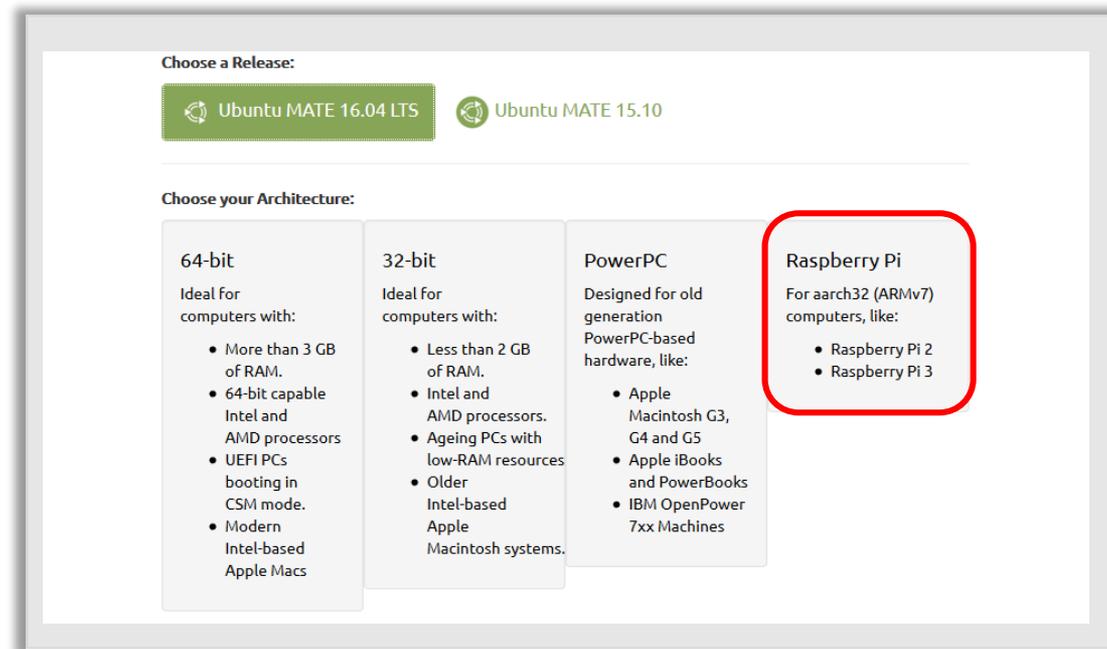


# Ubuntu MATE 16.04 LTSのイメージデータをダウンロード(1)

<https://ubuntu-mate.org/download/>



① Ubuntu MATE 16.04 LTSをクリック



② Raspberry Piをクリック

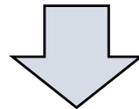
③ `ubuntu-mate-16.04-desktop-armhf-raspberry-pi.img.xz`をダウンロード

# ① 圧縮されたイメージファイルを解凍

7zip (<https://sevenzip.osdn.jp/>) を用いて解凍する

ubuntu-mate-16.04-desktop-armhf-raspberry-pi.img.xz

容量 約1.1 GB



ubuntu-mate-16.04-desktop-armhf-raspberry-pi.img

容量 約8 GB



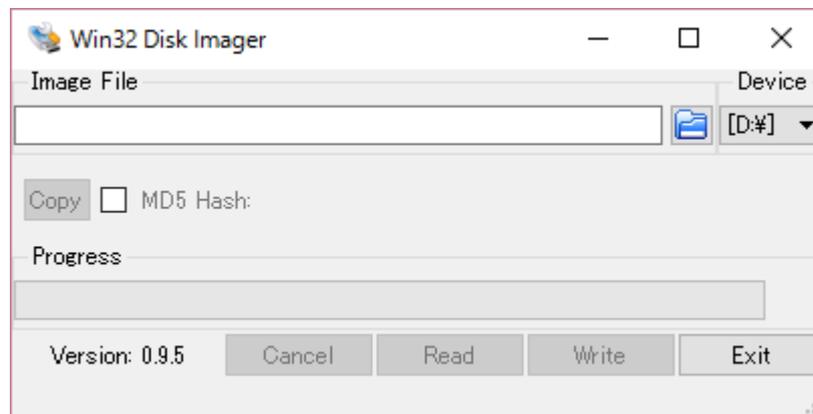
## ② イメージファイルをmicroSDカードに書き込む

Win32DiskImage ( <http://sourceforge.net/projects/win32diskimager/> )を用いて行う

### 注意

イメージファイルは8GBであるので、8GBのmicroSDカードがあれば通常は問題ない。しかしOpenFOAMをインストールするための容量が必要であることと、swap領域を確保する必要があるため、今回は**32GBのmicroSDカードを使用する**

(1) カードリーダーにmicroSDカードをセットし、Win32DiskImagerを起動する



(2) DeviceをmicroSDカードがマウントされているドライブを選択

(3)  ボタンをクリックし、ubuntu-mate-16.04-desktop-armhf-raspberry-pi.img  
を選択する

(4) Writeボタンをクリックする

microSDカード内のイメージデータ以外の部分は割り当てられていないので  
コマンドgparted (linux)を使用してパーティションの分割を行う

OpenFOAMのインストールのためにはswap領域を作成する必要があるので、  
構成は

Before

```
/dev/sda1 : boot 64MB  
/dev/sda2 : main 7.44GB
```

After

```
/dev/sda1 : boot 64MB(変更なし)  
/dev/sda2 : main 25.75GB (swap以外のすべて)  
/dev/sda3 : swap 4GB
```

とする

## gpartedコマンドを使用するための選択肢

- ① LinuxがインストールされているPCでgpartedをインストールし、使用する
- ② gpartedがインストールされているLiveCDを使用して、PCをCDブートで起動して使用する

gpartedコマンドを使用するための選択肢

① LinuxがインストールされているPCでgpartedをインストールし、使用する

linuxOS（今回はUbuntu MATE 16.04を使用）がインストールされているPCに、  
カードリーダーを用いてmicroSDカードを挿入する

端末を起動しgpartedを起動

無ければ

```
sudo apt-get install gparted
```

でインストールする

スライド17へ移動

## gpartedコマンドを使用するための選択肢

② gpartedがインストールされているLiveCDを使用して、PCをCDブートで起動して使用する

<http://gparted.org/download.php>

### Download Links

**Donate**

If you use GParted and it helps you with your task, then please consider making a donation to help us make GParted better.

**NOTE:** GParted Live is available in three different architectures

#### Architecture Description

**i686** 32-bit version runs on x86 and x86-64 based computers, limited to physical address space of 4 gigabytes, and uses one processor only.

*If in doubt, try this one first.*

**i686-PAE** 32-bit version runs on x86 (i686 and higher) and x86-64 based computers, with Physical Address Extension to access more than 4 gigabytes of memory, and supports use of multiple processor cores.

**amd64** 64-bit version runs on x86-64 based computers, with ability to access more than 4 gigabytes of memory, and supports use of multiple processor cores.

*For newer (>2010) computers with UEFI instead of legacy BIOS try this one.*

#### GParted Live CD/USB/HD/PXE Bootable Image

##### Stable Releases

[Download gparted-live-0.26.0-2-i686.iso](#) [Release notes](#) [Changelog](#) [Checksums](#)

[Stable directory \(.iso/.zip\)](#)  
(for i686, i686-pae and amd64 architectures)

##### Testing Releases

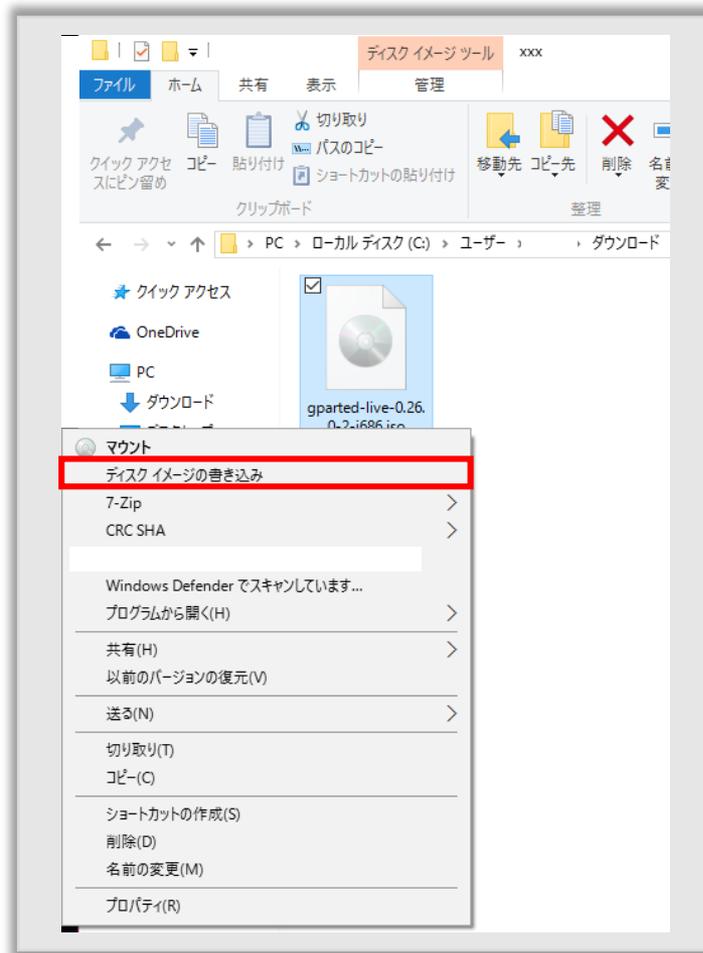
[Testing directory \(.iso/.zip\)](#)  
(for i686, i686-pae and amd64 architectures) Release notes and checksums in same directory.

gparted-live-0.26.0-2-i686.isoをダウンロード

# gparted-live-0.26.0-2-i686.isoをCD-Rに書き込む

Windows 10の場合

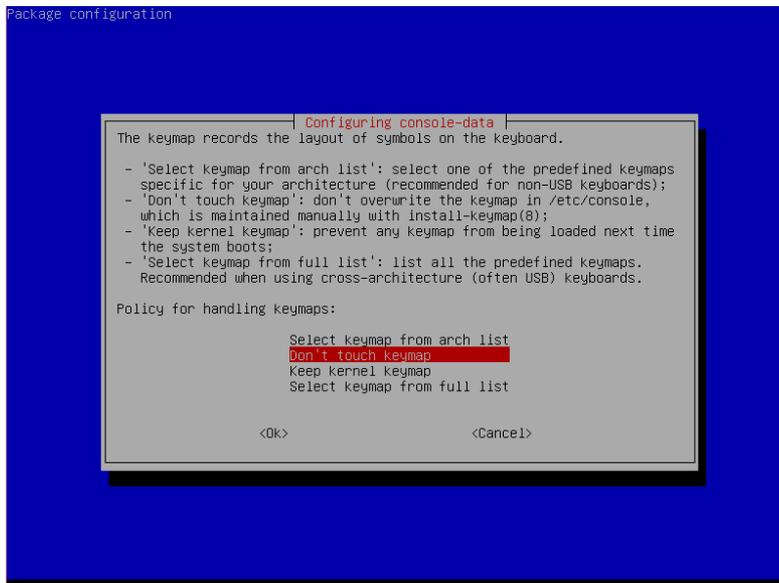
isoファイルを右クリック→[ディスクイメージの書き込み]をクリックでOK



## CD-Rをディスクドライブにセットし、再起動する



メニュー画面では  
GParted Liveを選択



Don't touch keymapを選択

```
Keep kernel keymap
Select keymap from full list

<OK>          <Cancel>

Looking for keymap to install:
NONE
*****
Loading language settings:
01: Portuguese (Brazilian) 18: Latvian
02: British English       19: Macedonian
03: Bulgarian            20: Norwegian
04: Catalan              21: Nepali
05: Czech                22: Portuguese
06: Dutch                 23: Punjabi
07: Finnish              24: Russian
08: French                25: Spanish
09: Galician             26: Simplified Chinese
10: German                27: Slovenian
11: Greek                 28: Swedish
12: Hebrew                29: Traditional Chinese (Hong Kong)
13: Hungarian            30: Traditional Chinese (Taiwan)
14: Italian               31: Turkish
15: Japan                 32: Ukrainian
16: Kinyarwanda           33: US English
17: Lithuanian            34: Vietnamese

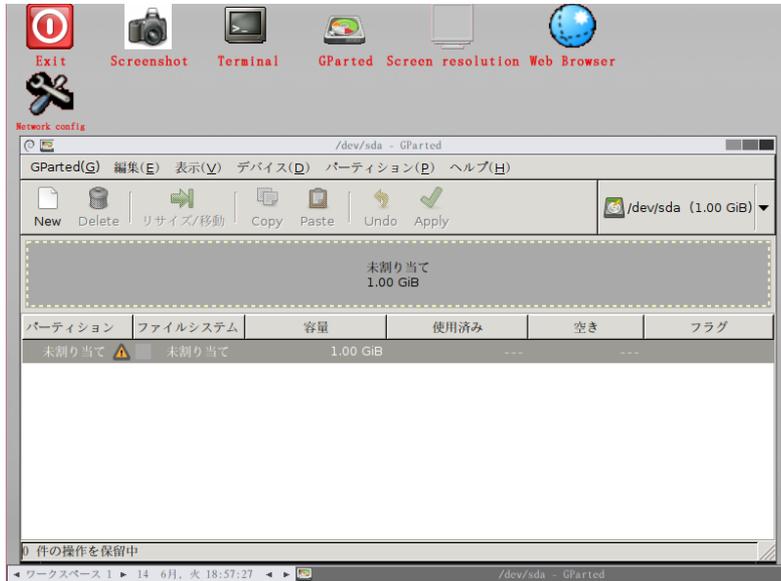
Which language do you prefer ?
[33] _
```

language選択は  
15(Japan)を選択

```
02: British English       19: Macedonian
03: Bulgarian            20: Norwegian
04: Catalan              21: Nepali
05: Czech                22: Portuguese
06: Dutch                 23: Punjabi
07: Finnish              24: Russian
08: French                25: Spanish
09: Galician             26: Simplified Chinese
10: German                27: Slovenian
11: Greek                 28: Swedish
12: Hebrew                29: Traditional Chinese (Hong Kong)
13: Hungarian            30: Traditional Chinese (Taiwan)
14: Italian               31: Turkish
15: Japan                 32: Ukrainian
16: Kinyarwanda           33: US English
17: Lithuanian            34: Vietnamese

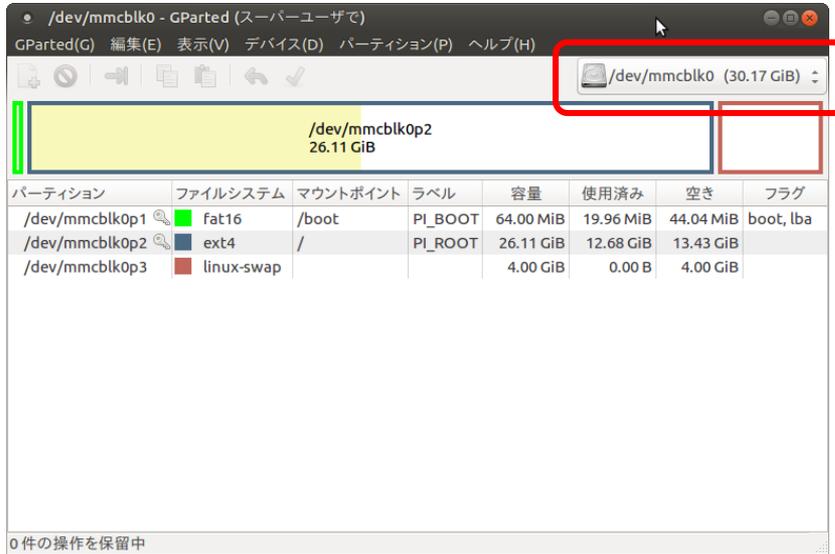
Which language do you prefer ?
[33] 15
Language selected ja_JP
Generating ja_JP locale by "localedef -f UTF-8 -i ja_JP ja_JP.UTF-8"... done!
Setting locale in /etc/default/locale...
done!
*****
//NOTE// Later we will enter graphical environment if you choose '0'. However, if graphical environment (X-window) fails to start, you can:
Run "sudo forcevideo" to configure it again. Choose 1024x768, 800x600 or 640x480 as your resolution, and the driver for your VGA card, etc. Most of the time you can accept the default values if you have no idea about them.
If failing to enter graphical environment, and it does not return to text mode, you can reboot again, and choose '1' here to config X manually.
-----
Which mode do you prefer ?
(0) Continue to start X to use GParted automatically
(1) Run 'forcevideo' to config X manually
(2) Enter command line prompt
[0]
```

そのままEnterキーを押す

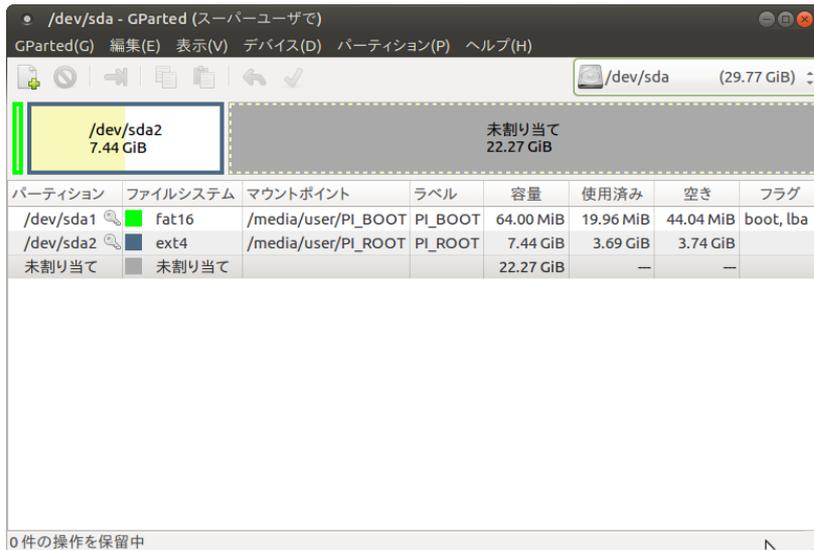


Gpartedが起動する

# Gpartedの使用方法



/dev/sda (microSDを選択aの文字は環境によって異なる)を選択



/dev/sdaのパーティション構成が表示される

パーティション アンマウントを選択

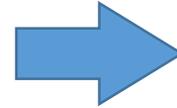
# /dev/sda2/領域を拡張する



The screenshot shows the GParted interface for /dev/sda. The /dev/sda2 partition is highlighted with a red box. Below the interface, the following text is displayed:

/dev/sda2/をクリックする  
右クリックでリサイズ/移動を選択

パーティション	ファイルシステム	マウントポイント	ラベル	容量	使用済み	空き	フラグ
/dev/sda1	fat16	/media/user/PI_BOOT	PI_BOOT	64.00 MiB	19.96 MiB	44.04 MiB	boot, lba
/dev/sda2	ext4	/media/user/PI_ROOT	PI_ROOT	7.44 GiB	3.69 GiB	3.74 GiB	
未割り当て	未割り当て			22.27 GiB	-	-	



The dialog box shows the resize/move options for /dev/sda2. The '新しいサイズ (MiB):' field is set to 7615.

最小サイズ: 3783 MiB    最大サイズ: 30415 MiB

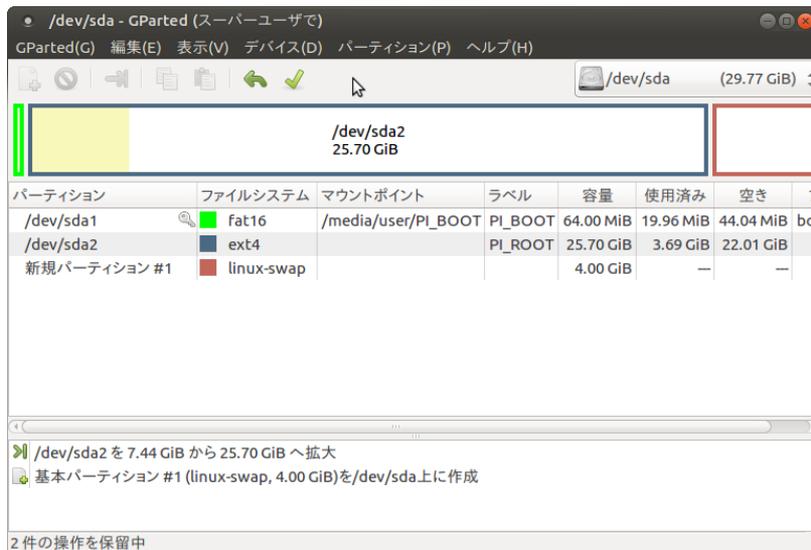
前方の空き領域 (MiB): 0

新しいサイズ (MiB): 7615

後方の空き領域 (MiB): 22800

位置あわせ: MiB

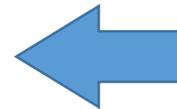
キャンセル(C)    リサイズ/移動



The screenshot shows the GParted interface after the expansion. The /dev/sda2 partition is now 25.70 GiB. A new swap partition is also visible. Below the interface, the following text is displayed:

/dev/sda2を7.44 GiBから25.70 GiBへ拡大  
基本パーティション #1 (linux-swap, 4.00 GiB)を/dev/sda上に作成

パーティション	ファイルシステム	マウントポイント	ラベル	容量	使用済み	空き	フ
/dev/sda1	fat16	/media/user/PI_BOOT	PI_BOOT	64.00 MiB	19.96 MiB	44.04 MiB	boo
/dev/sda2	ext4		PI_ROOT	25.70 GiB	3.69 GiB	22.01 GiB	
新規パーティション #1	linux-swap			4.00 GiB	-	-	



The dialog box shows the final resize/move options for /dev/sda2. The '新しいサイズ (MiB):' field is set to 26319.

最小サイズ: 3783 MiB    最大サイズ: 30415 MiB

前方の空き領域 (MiB): 0

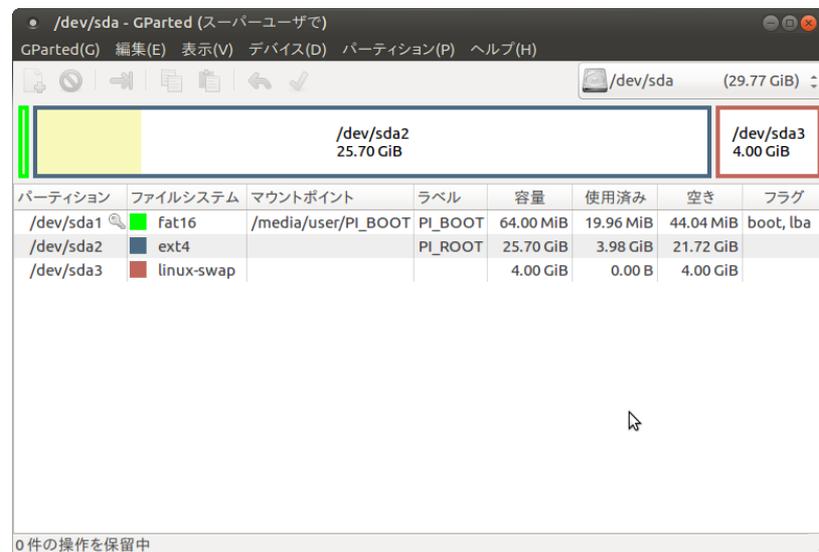
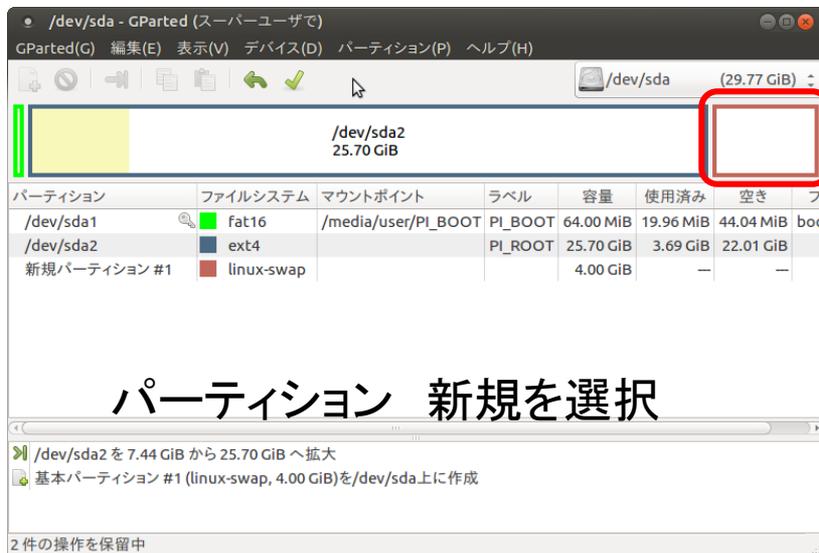
新しいサイズ (MiB): 26319

後方の空き領域 (MiB): 4096

位置あわせ: MiB

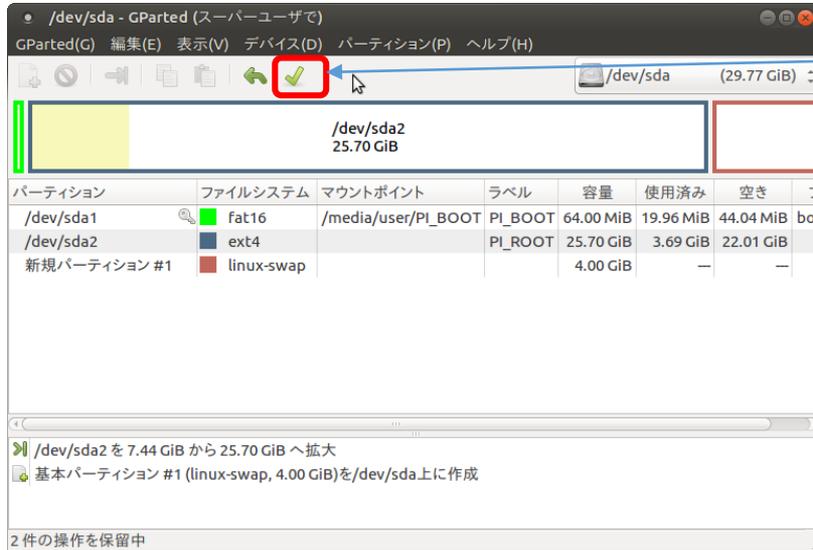
キャンセル(C)    リサイズ/移動

# スワップ領域を設定する

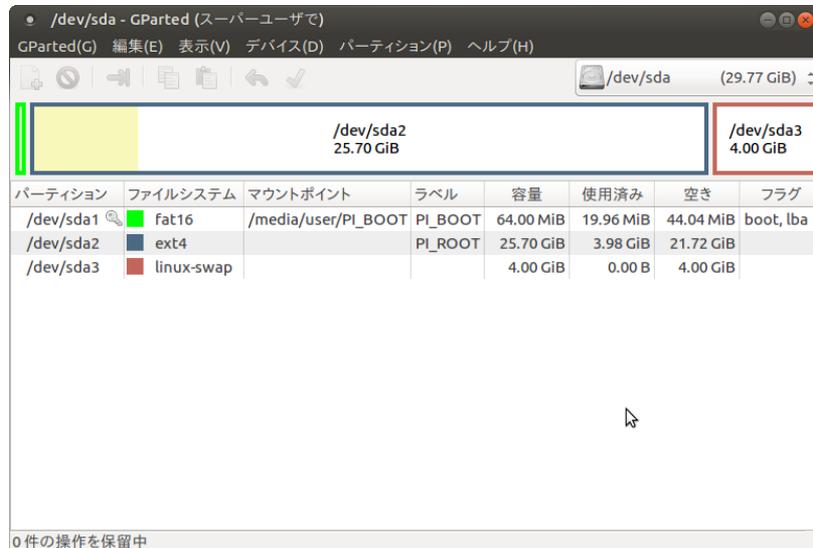


ファイルシステム linux-swapを選択

# 設定したパーティションを作成する



適用ボタンをクリックする



## Raspberry piにmicroSDカードを挿入し起動する

スワップを割り当てる

```
sudo mkswap /dev/mmcblk0p3↵  
sudo swapon /dev/mmcblk0p3↵
```

スワップ領域の自動マウント

/etc/fstabを以下のように編集する

proc	/proc	proc	defaults	0	0
/dev/mmcblk0p2	/	ext4	defaults,noatime	0	1
/dev/mmcblk0p1	/boot/	vfat	defaults,noatime	0	2
/dev/mmcblk0p3	swap	swap	defaults,noatime	0	0

再起動する

02

## Raspberry PiにOpenFOAMをインストール

## OpenFOAM Wikiの情報を参考にインストールを行う

[https://openfoamwiki.net/index.php/Installation/Linux/OpenFOAM-3.0.1/Ubuntu#Ubuntu\\_16.04](https://openfoamwiki.net/index.php/Installation/Linux/OpenFOAM-3.0.1/Ubuntu#Ubuntu_16.04)

### 必要なパッケージをインストール

```
sudo apt-get install build-essential binutils-dev cmake flex bison zlib1g-dev qt4-dev-tools libqt4-dev libqtwebkit-dev gnuplot ¥  
libreadline-dev libncurses-dev libxt-dev libopenmpi-dev openmpi-bin libboost-system-dev libboost-thread-dev libgmp-dev ¥  
libmpfr-dev python python-dev libcglib-dev libglu1-mesa-dev libqt4-opengl-dev
```

インストール時間: 約15分

### ソースファイルのダウンロードとファイルの展開

```
cd ~  
mkdir OpenFOAM  
cd OpenFOAM  
wget "http://downloads.sourceforge.net/foam/OpenFOAM-3.0.1.tgz?use_mirror=mesh" -O OpenFOAM-3.0.1.tgz  
wget "http://downloads.sourceforge.net/foam/ThirdParty-3.0.1.tgz?use_mirror=mesh" -O ThirdParty-3.0.1.tgz  
  
tar -xzf OpenFOAM-3.0.1.tgz  
tar -xzf ThirdParty-3.0.1.tgz
```

## RaspberryPi用にソースファイルの編集

### ファイル

OpenFOAM/ThirdParty-3.0.1/etc/wmakeFiles/scotch/Makefile.inc.i686\_pc\_linux2.shlib-OpenFOAM-32Int32

から“-m32”の記述を取り除く

### ファイル

OpenFOAM/OpenFOAM-3.0.1/wmake/rules/linuxARM7Gcc

を以下のように編集

edit file: cOpt

move the # from 1st to 2nd line:

```
cOPT = -O3 -floop-optimize -falign-loops -falign-labels -falign-functions -falign-jumps -fprefetch-loop-arrays -mcpu=cortex-a9 -mfpu=vfpv3-d16 -mfloat-abi=hard  
#cOPT = -O3 -floop-optimize -falign-loops -falign-labels -falign-functions -falign-jumps -fprefetch-loop-arrays -mcpu=cortex-a9 -mfpu=vfpv3-d16 -mfloat-abi=softfp
```

edit file: c++Opt

```
c++OPT = -O3 -floop-optimize -falign-loops -falign-labels -falign-functions -falign-jumps -fprefetch-loop-arrays -mcpu=cortex-a9 -mfpu=vfpv3-d16 -mfloat-abi=hard  
#c++OPT = -O3 -floop-optimize -falign-loops -falign-labels -falign-functions -falign-jumps -fprefetch-loop-arrays -mcpu=cortex-a9 -mfpu=vfpv3-d16 -mfloat-abi=softfp
```

## OpenFOAMのインストールのための設定を行う

### MPIを使用する設定

```
In -s /usr/bin/mpicc.openmpi OpenFOAM-3.0.1/bin/mpicc ↵  
In -s /usr/bin/mpirun.openmpi OpenFOAM-3.0.1/bin/mpirun ↵
```

### CGAL 4.7を使用するように以下のコマンドを実行する

```
sed -i -e 's/^¥(cgal_version=¥).*/¥1cgal-system/' OpenFOAM-3.0.1/etc/config/CGAL.sh ↵
```

### 環境変数を定義

```
source $HOME/OpenFOAM/OpenFOAM-3.0.1/etc/bashrc WM_ARCH_OPTION=32 ↵
```

## OpenFOAMのインストール

```
cd $WM_PROJECT_DIR ↵  
./Allwmake -j 4 > log.make 2>&1 ↵
```

↑  
並列数

← log.makeに標準出力とエラー出力  
が書き込まれる

インストール時間目安: 約11時間

## 設定は正しいはずなのにエラーが出る場合は

log.makeの51行目に

Boost does not appear to be installedと表示されているかを確認する

もしあれば

ThirdParty-3.0.1/platforms/linuxARM7Gcc/boost-systems

ThirdParty-3.0.1/platforms/linuxARM7Gcc/cgal-systems

という空のディレクトリが作成されているので消去する

消去後 ./Allwmakeを実行する

log.makeの55行目に

BOOST\_VERSION\_NO=105800と表示されていればOK

## paraviewインストール

```
cd $WWM_THIRD_PARTY_DIR
export QT_SELECT=qt4
./makeParaView4 -python -mpi -python-lib /usr/lib/arm-linux-gnueabi/libpython2.7.so.1.0 > log.makePV
```

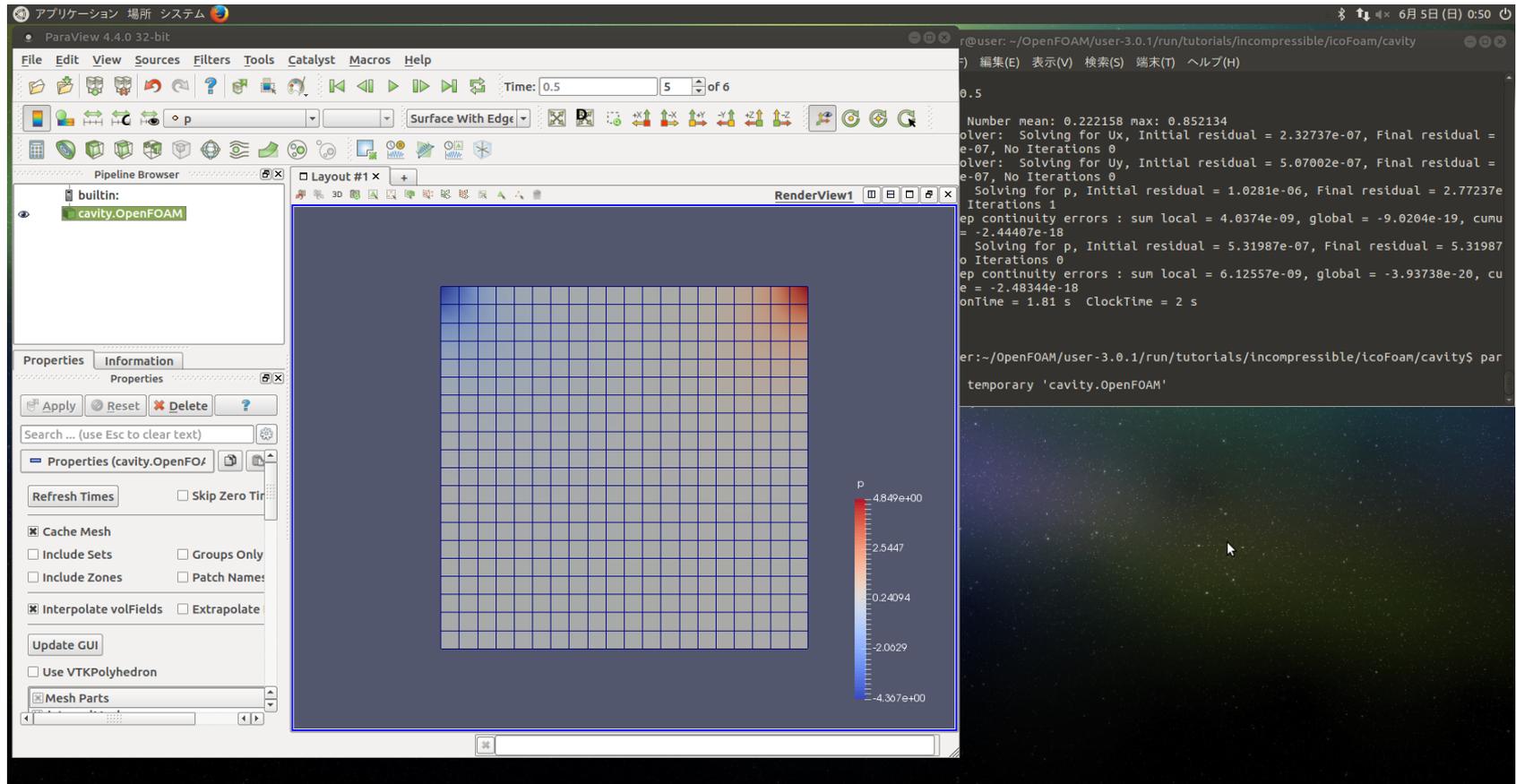
インストール時間目安: 7時間30分

## paraview reader module libraries インストール

現在の状態ではparaFoamを実行できないので以下を実行する

```
cd $FOAM_UTILITIES/postProcessing/graphics/PV4Readers
./Allwclean
./Allwmake
```

# 丸1日のインストール作業の末.....



OpenFOAMの環境構築できました！！