

# 30分でできる！ Linux超入門

2018年3月15日

KMC 1 回 polaris

 [hide4d51@gmail.com](mailto:hide4d51@gmail.com)

# 自己紹介

# 自己紹介

## **polaris**

- KMC1回生(次の2回生)
- 京都工芸繊維大学 情報工学課程 1回
- C、Python、電子工作、お絵かきを少し等
- 変な世界観の電子ゲームが好き

# 自己紹介(Continued)

- <https://h-teramura.github.io/>  
→日記や知見を書いたりするところ(ほぼ更新していない)
- Slack #polaris-memo  
→観察したい人はどうぞ
- このアイコンの人です



# 今日の流れ

- とにかくインストールしてみる
- Linuxとは何者か？
- 基本的な操作方法
- CLI
- コマンドいろいろ
- 超入門のその先
- 質問会

さっそくですが

いまから**Linux**を使う環境を作るので  
**PC**を開いてね

# 仮想化ソフトのインストール

# 仮想化ソフト

- PCの上にPCを再現するソフト
- 再現したPCを「仮想機械(マシン)」と呼ぶ
- リソースが許す限りPCを作り出せる
- ネットワークやデバイス(マウスとかのおもにUSB接続のもの)を仮想マシンに使わせることもできる

→これを使うことで**PC**を買うことなく**Linux**環境が作れる



# 仮想化ソフト(Continued)

有名な4つを紹介

- QEMU
- Microsoft Hyper-V
- VMware Workstation Player
- Oracle VM VirtualBox

今回は**VirtualBox**を使います

# マシンのスペック

仮想マシンを動かす場合はそれなりに性能が良いPCがいる

今回は

- 2つ以上のコアを持つ64bitCPU
- 4GB以上のRAM(2GBでもいけるかな...?)
- 20GB以上の空き領域を持つHDD/SSD
- Wi-Fi  
を搭載したマシンを想定している

# そんなPCないよーという場合

オンラインのを使いましょう

- Webminal(<http://www.webminal.org/>)  
ユーザー登録がいる
- JSLinux(<https://bellard.org/jslinux/>)

ほかにもLinuxの機能はないけどシェルの操作を練習できるサイトもある

# ダウンロード

インストールしている間に

# 実際のLinuxのセットアップ 方法

(今回はやらない)

# インストールディスクイメージのダウンロード

- 「Linuxディストリビューション」とかで検索してよさそうなものをダウンロードしてくる
- 有名なのは「Ubuntu」「Debian」「Fedora」など

## 概要 [編集]

Linuxディストリビューションについての基本的な一般情報：製作者、開発元、リリース日、最新版など。

| ディストリビューション                                      | 製作者   | 開発元   | 最初の公式リリース<br>(年-月-日) | ベースディストリ<br>ビューション                               | 最新安定版バー<br>ジョン                  | リリース日                                  | 用途                              |
|--|---|---|----------------------|--|---------------------------------|--|---------------------------------|
| <a href="#">aptosid</a>                          | <a href="#">Stefan Lippers-Hollmann (slh)</a>   | <i>aptosid</i> チーム  | 2010-09-14           | <a href="#">Debian Sid</a>                       | 2013-01<br>( <i>Εσπερίδες</i> ) | 2013-05-05                             | 汎用                              |
| <a href="#">Arch Linux</a>                       | <a href="#">Aaron Griffin</a>   | <i>dev</i> チーム  | 2002-03-11           | none   | 2017.05.01                      | 2017-05-01, <a href="#">ローリング・リリース</a> | 汎用                              |
| <a href="#">Asianux</a>                          | <a href="#">Red Flag Linux</a> , <a href="#">Miracle Linux</a> , <a href="#">Haansoft Linux</a> | <i>dev</i> チーム  | 2004-06              | <a href="#">RHEL</a>                             | 4.0                             | 2011-07-18                             | ワークステーション, サーバー                 |
| <a href="#">Bash on Ubuntu on Windows</a>        | <a href="#">Canonical Ltd<sup>[1]</sup></a> , <a href="#">Microsoft Corporation</a>             | <a href="#">Canonical Ltd</a> , <a href="#">Microsoft Corporation</a> | 2016-08-03           | <a href="#">Ubuntu</a>                           | Windows 10<br>Creators Update   | 2017-04-11                             | デスクトップ                          |
| <a href="#">CentOS</a>                           | <a href="#">CentOS Project</a>  | <a href="#">CentOS Project</a>  | 2003-12              | <a href="#">RHEL</a>                             | 7.1511                          | 2015-12-14                             | サーバー, ワークステーション, デスクトップ, ラップトップ |
| <a href="#">Damn Small Linux</a>                 | <a href="#">ジョンアンドリュース</a>  | <i>dev</i> チーム  | 2003                 | <a href="#">KNOPPIX</a>                          | 4.4.10                          | 2008-11-17                             | 汎用                              |
| <a href="#">Debian GNU/Linux</a>                 | <a href="#">Ian Murdock</a>   | <a href="#">Debian Project</a>  | 1993-08-16           | none   | 9.1 ( <i>stretch</i> )          | 2017-07-22                             | 汎用                              |
| <a href="#">Fedora</a>                           | <a href="#">Fedora Project</a>  | <a href="#">Fedora Project</a>  | 2003-11-05           | <a href="#">Red Hat Linux</a>                    | 24                              | 2016-06-21                             | 汎用                              |
| <a href="#">Foresight Linux</a>                  | <a href="#">Ken VanDine</a>   | <i>dev</i> チーム  | 2004-12              | <a href="#">rPath</a>                            | 2.5.3                           | 2013-04-12, <a href="#">ローリング・リリース</a> | デスクトップ, ワークステーション               |
| <a href="#">Gentoo Linux</a>                     | <a href="#">Daniel Robbins</a>  | <a href="#">Gentoo Foundation, Inc.</a>                               | 2002-03-31           | none   | stable                          | 2016-02-29, <a href="#">ローリング・リリース</a> | 汎用                              |
| <a href="#">Kanotix</a>                          | <a href="#">Jörg Schirottke (Kano)</a>  | <i>dev</i> チーム  | 2003-12-24           | <a href="#">Debian</a> , <a href="#">KNOPPIX</a> | 2014                            | 2014-05-10                             | Live CD, デスクトップ                 |
| <a href="#">KNOPPIX</a>                          | <a href="#">Klaus Knopper</a>   | <i>dev</i> チーム  | 2003-01-19 (3.1)     | <a href="#">Debian</a>                           | 7.6.1                           | 2016-01-25                             | Live CD                         |
| <a href="#">Linspire</a>                         | <a href="#">Lindows.com, Inc.</a>   | <a href="#">Linspire, Inc.</a>  | 2002-03              | <a href="#">Debian</a>                           | 6.0                             | 2007-10-10                             | デスクトップ                          |
| <a href="#">Linux Mint</a>                       | <a href="#">Clement Lefebvre</a>  | <i>dev</i> チーム  | 2006-08-27           | <a href="#">Ubuntu</a>                           | 18.3                            | 2017-11-27                             | デスクトップ                          |
| <a href="#">LMDE (Linux Mint Debian Edition)</a> | <a href="#">Clement Lefebvre</a>  | <i>dev</i> チーム  | 2010-09-07           | <a href="#">Debian</a>                           | LMDE2                           | 2015-04-10                             | デスクトップ                          |

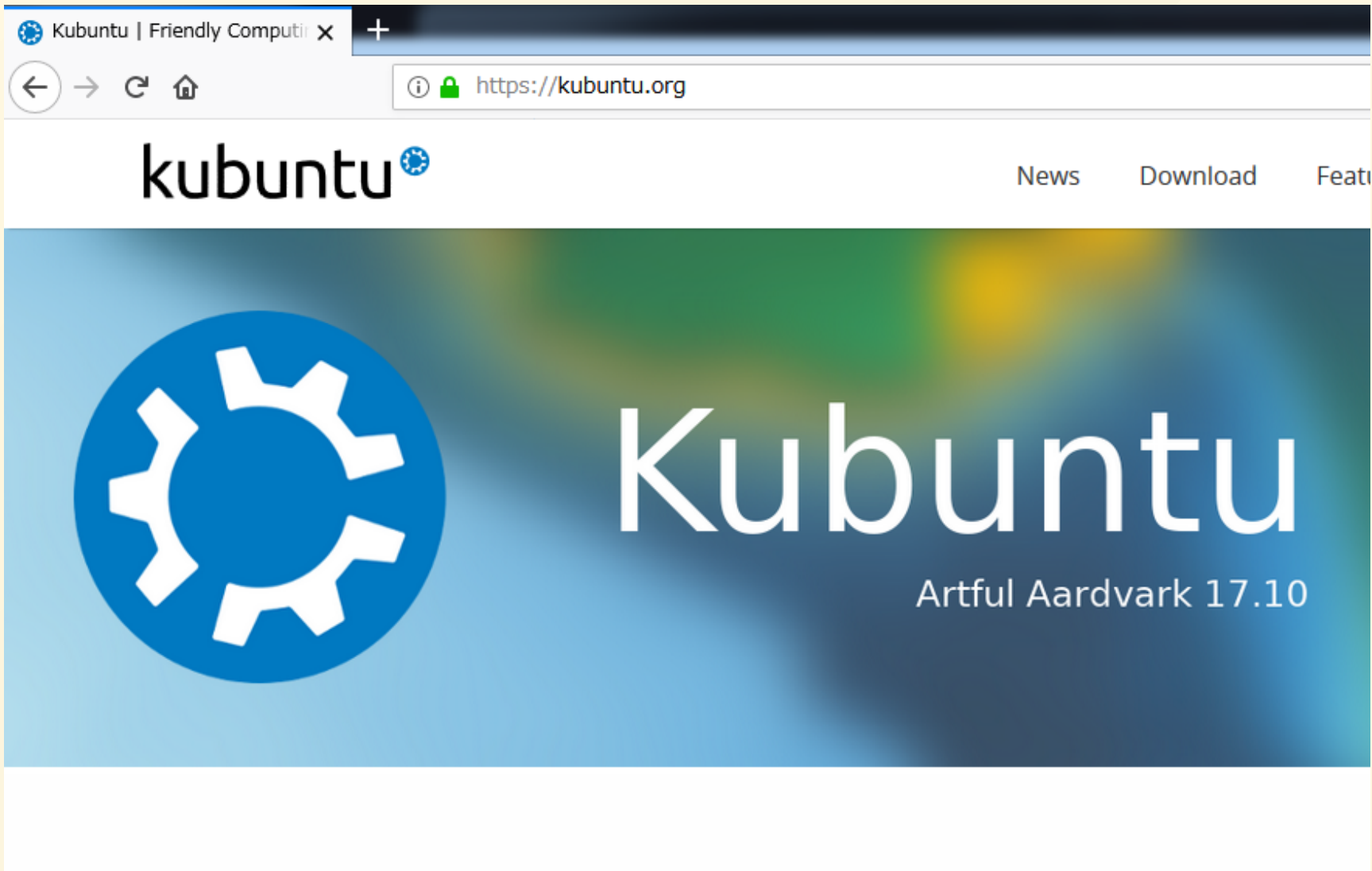
# 多い！

用途や動かすマシンの性能、ユーザの熟練度に合わせていろいろなものが開発されている

→こんなディストリビューションが良いぞ、みたいなまとめサイトなどを利用しがんばって探す

たとえば**kubuntu**をインストールする場合





Kubuntu | Friendly Computi x +

← → ↻ 🏠 <https://kubuntu.org>

kubuntu  News [Download](#) Feat



 **Kubuntu**  
Artful Aardvark 17.10

これ

The image shows a screenshot of the Kubuntu website. The browser's address bar displays 'https://kubuntu.org'. The website's navigation menu includes 'News', 'Download', and 'Feat'. The 'Download' link is highlighted with a red box, and a red arrow points to it from a white box containing the Japanese characters 'これ' (kore), which means 'this' or 'here'. The main banner features the Kubuntu gear logo and the text 'Kubuntu Artful Aardvark 17.10'.

## Kubuntu 16.04.4 LTS

The Long Term Support (LTS) version of the Kubuntu operating system for desktop PCs and laptops, Kubuntu 16.04 LTS supported with security and maintenance updates, until April 2019.

**Recommended for most users.**

[Kubuntu 16.04 LTS release notes](#)

Choose your download (64-bit Recommended)

**64-bit Download**

**32-bit Download**

[Alternative downloads, torrents, mirrors and check-sums](#)

- 最近は64ビット版を選べばたいてい何とかなる  
(そもそも一部の32bitマシンはもうサポートされていない(i386系CPUなど))
- 自動的に最も速いサーバからダウンロードされることが多い(されないこともある)

# イメージが死んでないかチェック

- やらなくてもたいてい問題ないが検査することがすすめられている

## Checksums

Before writing an image to DVD or USB drive, it is highly recommended that you verify the SHA256 sum (hash) of the ISO file. For instructions, please see [HowToSHA256SUM](#). Below is a list of SHA256 sums to check with your downloaded file.

---

**kubuntu-17.10.1-desktop-amd64.iso** - *1224c233ed014cefe824a289a9e6d7ee446fb79bb1d2bebdd55c87d0ec4feced*  
**kubuntu-17.10.1-desktop-i386.iso** - *08402799f5c7512fd87a9ec84f7c94d5d70b02a8341efb2e73a90a18076a1205*

---

**kubuntu-16.04.4-desktop-amd64.iso** - *9e4b9c0fcd511740606b67edbcde0ae83adce3bdabbb1ec9706cc6545adb90a5*  
**kubuntu-16.04.4-desktop-i386.iso** - *c82bcbb0432937c961c6747a6d206b2e962fda53498c77b1d861a0f6ef750b22*

---

**kubuntu-14.04.5-desktop-amd64.iso** - *04c32d9823b2fa724fbd675a2c90c5e50211cb1745a451db0d1616189df228e2*  
**kubuntu-14.04.5-desktop-i386.iso** - *9cf450a0d4049ca504a765afa5be44955702d4b54f59bb81ec89d18dfc96fd0c*

SHA256で計算すればよいらしいのでGit Bashで

```
$ sha256sum kubuntu-16.04.4-desktop-amd64.iso
```

```
9e4b9c0fcd511740606b67edbcde0ae83adce3bdabbb1ec9706cc6545  
adb90a5 *kubuntu-16.04.4-desktop-amd64.iso
```

(ほかのツールでやっても良い)

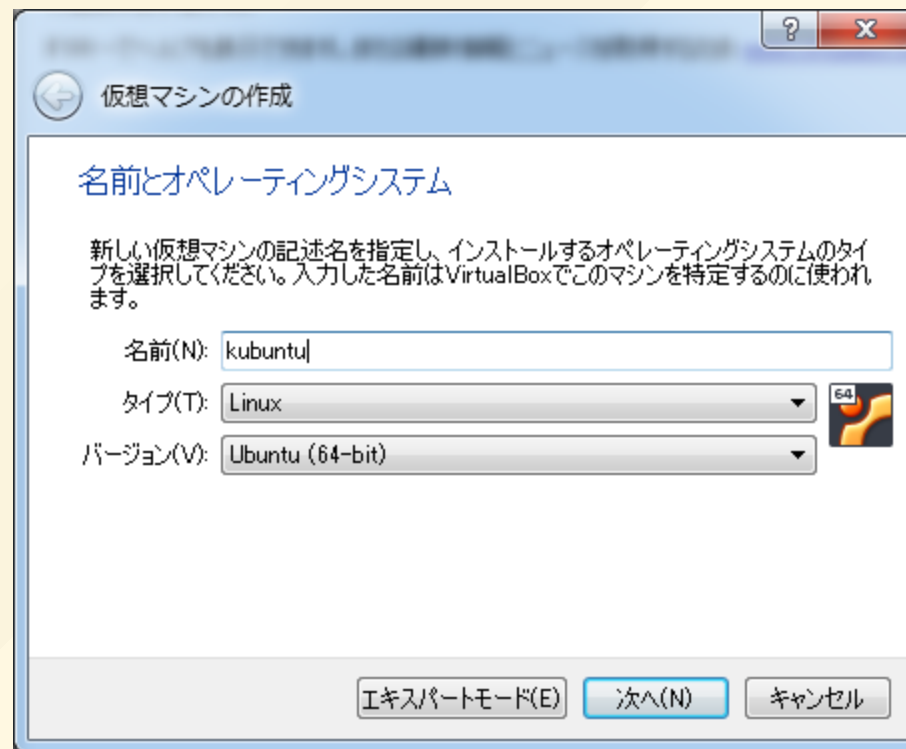
# 仮想マシンの作成

VirtualBoxに仮想マシンを構築する

- ダウンロードしたイメージを渡すといい感じにしてくれる
- 便利

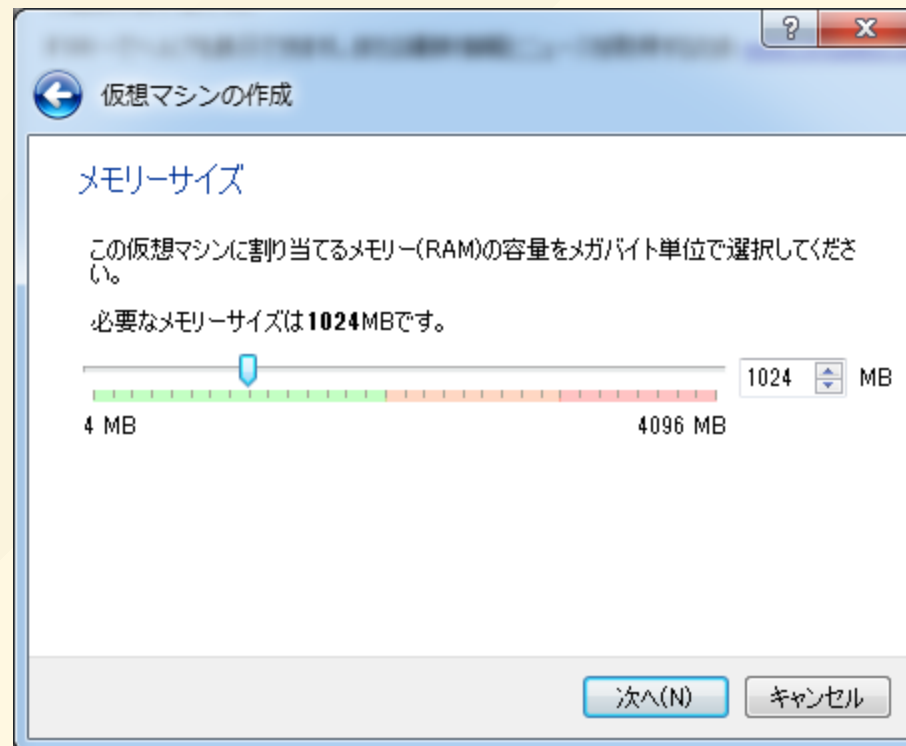
# 仮想マシンの作成(Continued)

- VirtualBoxを起動して、「新規」をクリック  
→ 「仮想マシンの作成」ウィザードが表示される



# 仮想マシンの作成(Continued)

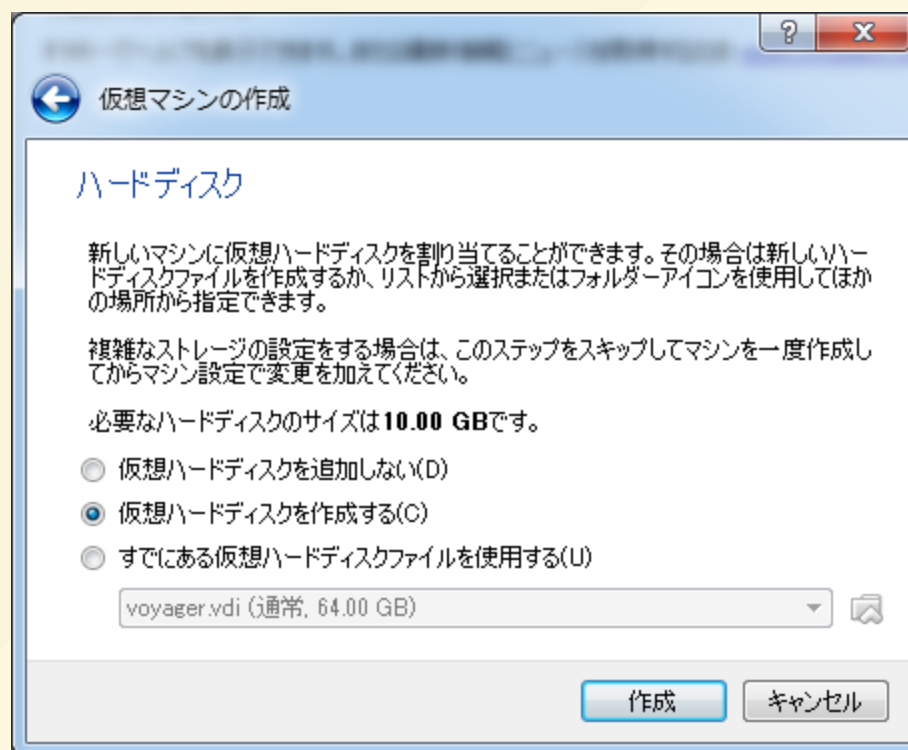
- 仮想マシンのメモリ容量を決定する  
→重い処理をやらせるならデフォルトより大きくする





# 仮想マシンの作成(Continued)

- 仮想ハードディスクの作成  
→ここは仮想ハードディスクを作成を選べばOK



# 仮想マシンの作成(Continued)

ハードディスクのファイルタイプはVDI  
物理ハードディスクにあるストレージで可変サイズ  
ーを選ぶ

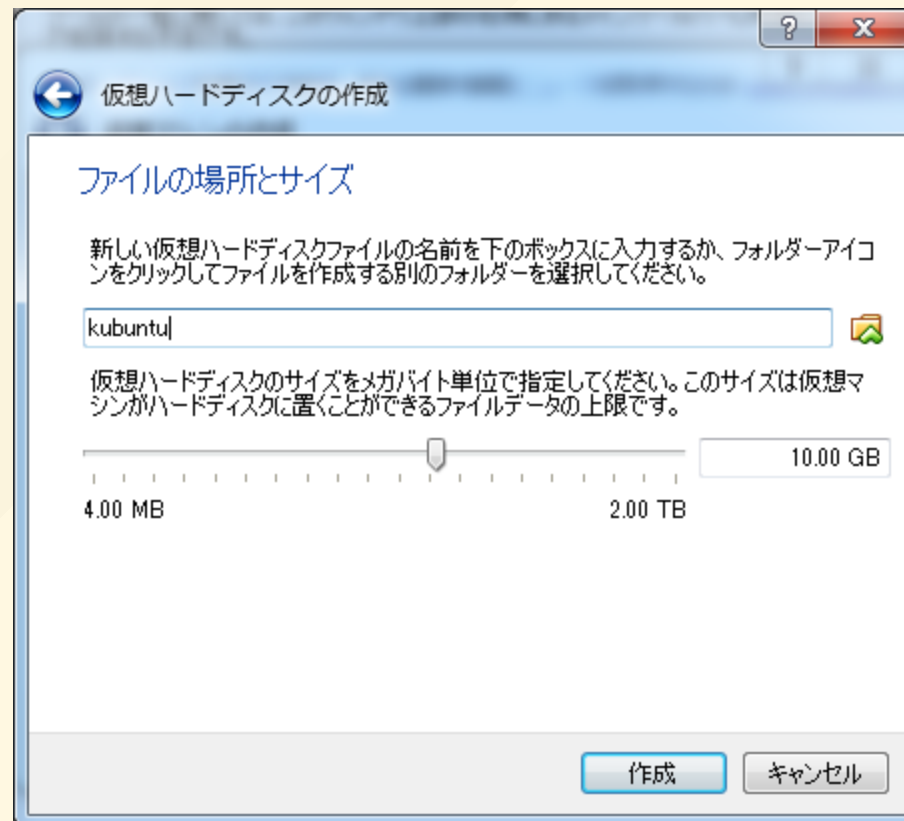
ちなみに

- VHD: Microsoft Hyper-V
- VMDK: VMware Workstation Player  
の仮想ディスク形式

可変サイズにすると使っただけサイズが増えていく  
(速度に少しオーバーヘッドがあるかもしれない)

# 仮想マシンの作成(Continued)

- 保存フォルダ名は仮想マシン名(デフォルト)にしておくとわかりやすい
- サイズはほしいだけのサイズにする



# 仮想マシンの設定

ここまできたら作成した仮想マシンを一回クリックで選んで「設定」を押す(まだ起動しないように、動かないので)

# 仮想マシンの設定(Continued)

これだけはやっておこう

- システム>プロセッサ>プロセッサ数を上げる  
仮想マシンのパフォーマンスが良くなる
- システム>プロセッサ>PAE/NXを有効化をON  
メモリ管理の拡張らしい。ないと動かないOSがある(気がする)
- ストレージ>コントローラ:IDEのCDのアイコン  
これを選んで属性のところにあるCDのアイコンをクリック。仮想光学ディスクファイルを選択を選んでさっきダウンロードしたイメージを開く

# 仮想マシンの起動

- 設定画面をOKをクリックして閉じ、起動をクリック

→先ほど開いたイメージから起動する

ここからの作業はディストリビューション毎に異なるのでググるなり人に聞くなりすることになる

というわけで

セットアップ済みの環境を  
お配りします



# ダウンロード

# 仮想マシンのインポート

- 「ファイル」 > 「仮想アプライアンスのインポート」をクリック

インポートしている間に

# Linuxとは何者か？

- オペレーティングシステム(Operating System)の一種  
→WindowsやmacOSもその仲間
- Linuxというとオペレーティングシステムの中核をさすことが多い

# オペレーティングシステムの中核

- カーネル(Kernel)と呼ばれるもの
- 主に計算機の初期化とタスク(やりたいこと)、計算資源の管理を行うプログラム
- アプリケーションを実行するのに必要なことをすべて担っているわけではない  
→実際、GUIを使うにはそれを管理するアプリケーションがいる(Xサーバ等)

# Windows/macOSは？

- これらにもカーネル(のようなもの)はある
- カーネルの助けを借りてさまざまなアプリケーションが実行され、今見ている風になっている

# デーモン(Daemon)

- カーネルのほかにバックグラウンドで実行されて通信などの機能を提供するプログラムのこと
- UNIXとそれに近いOSの界隈で使われる用語  
→Windowsでいう「サービス」

たとえば.....

- sshd: SSH通信を取り持つ
- httpd: HTTPサーバ(Apache)
- systemd: デーモンを起動するデーモン

# Linuxディストリビューション

- 調整されたLinuxカーネルとデーモンやよく使うプログラムをひとまとめにしたソフトウェアバンドル
- ビルド済みのアプリケーション(パッケージ)を配布していたり、商用利用を前提としたサポートをやっていることもある

# Debian系

- Debian Linuxとそれを元に作られたディストリビューション
- よく使われる
- プロプライエタリなソフトウェアがサポートしていることが多い(私見)

## 例

- Ubuntu(Kubuntu,Xubuntu,Lubuntu, etc.)
- Linux Mint



# Red Hat系

- Red Hat Enterprise Linuxとそれを元に作られたディストリビューション
- サーバに使ったりする
- RHELはライセンスを買うとRed Hatに電話サポートしてもらえる(一般人向けではない)
- パッケージが古くなりがち(CentOSなど)

## 例

- Fedora
- CentOS

# その他有名なもの

- Gentoo(じえんとうー) Linux  
ソースコードをビルドして環境構築する
- openSUSE(すーぜ)、SLES  
SUSEとコミュニティが開発している。サーバ、ワークステーションなど
- Arch Linux  
polarisが好きなディストリビューション
- Slackware

# 実際に使う上で気になる違い

- Linuxではカーネルと重要なプログラムがはっきり分かれた状態で提供される  
→LinuxではWindows ExplorerやFinderのような比較的重要なプログラムでも好みのものに差し替えることができる
- 文字コード、改行文字の違い
  - Windows: Shift-JIS, CR+LF
  - macOS: ??? (いろいろな情報がある), LF
  - Linux: (たいてい、変更可能)UTF-8, LF

# 実際に使う上で気になる違い (Continued)

- Silverlightは動かない  
→ほかにもLinux版のAdobe Flash Playerのサポートが終わっているなど
- ファイルの扱いの違い(後述)

# 基本的な操作方法

# ログインしてみる

- 仮想マシンを起動
- ログイン画面が出てきたらパスワードを入力して"Login"をクリック or Enterキーを押す

デフォルトの設定(polarisが設定しました)

- ユーザー名: springcamp
- パスワード: gogo575mogamigawa

# たいていGUI系はついてくる

- 操作方法は大体Windows/macOSのそれと同じ  
→この仮想マシンでは
  - デスクトップマネージャ"LightDM"
  - デスクトップ環境"LXDE"
  - その他GUI採用アプリ
- ファイル操作やパッケージのアップグレード・インストールも大体できてしまう  
→多分「ソフトウェアの更新」なんていう画面が出てるはず

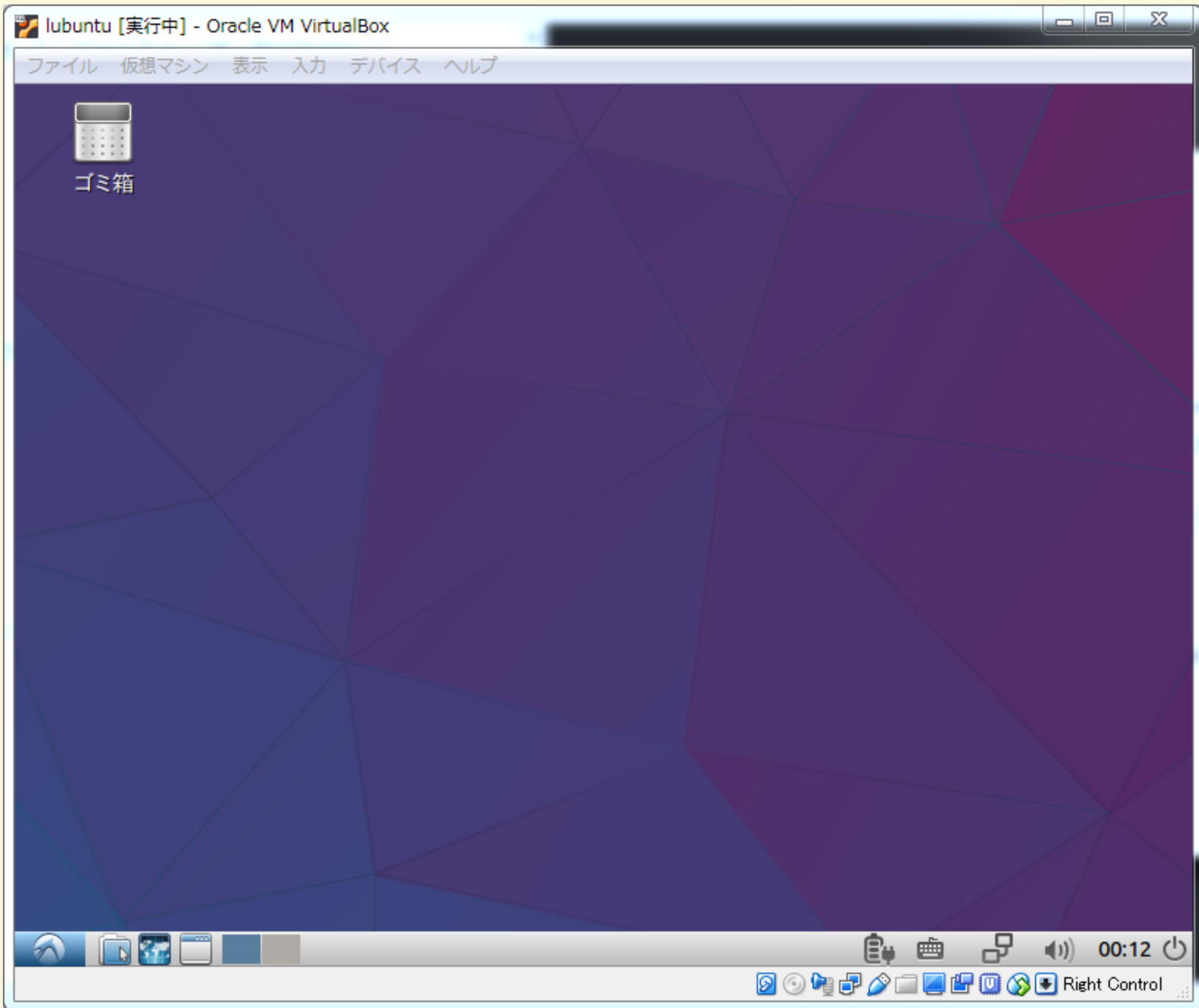
# たいていGUI系はついてくる (Continued)

- デスクトップマネージャ  
ログイン画面の表示を行う  
GDM,SDDM,LightDMなど
- デスクトップ環境  
GUI環境を実現するためのソフトウェア群  
KDE Plasma,GNOME,LXDE,Unityなど



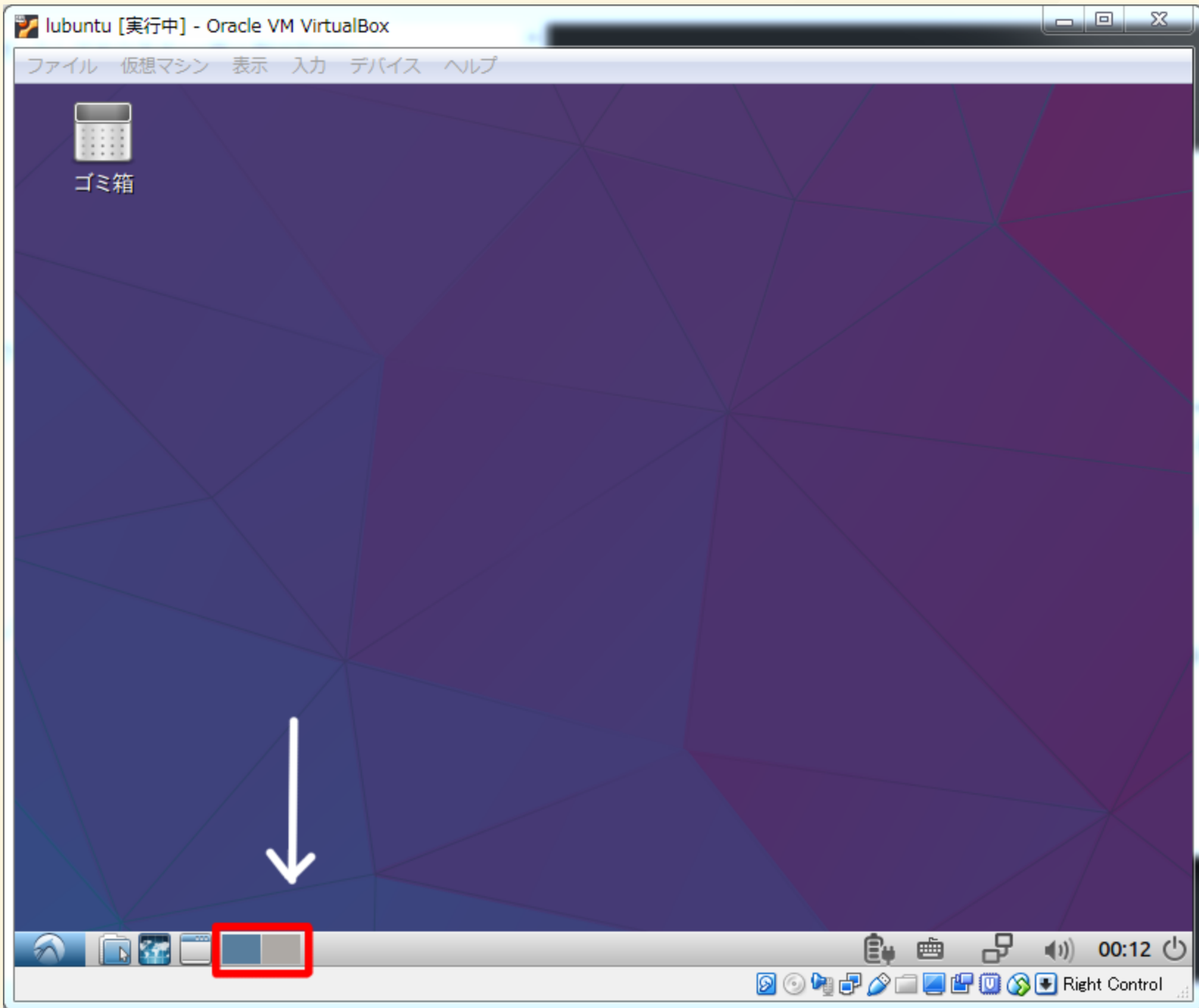
# LXDE

- 軽量なデスクトップ環境
- 初期設定ではWindowsに似ている(私見)



# 仮想デスクトップ

- macOSみたいに複数のデスクトップを扱える  
→1番目にブラウザ、2番目にIDEとターミナルなど
- Ctrl+Alt+左右で切り替え
- Logoキー(Windows,Command)+F1～で番号指定切り替え
- あるいはマウスでパネル上のウィジェットをクリック



**CLI**

# CLI

- CommandLine Interfaceの略
- コマンドを入力することで操作するようなインターフェース
- CUIがほぼ同義な語として使われることがある

こんな感じ:

```
$ tar -cJf pictures.tar.xz pictures
$ scp ./pictures.tar.xz polaris@kmc.gr.jp:~/downloads/
.....
```

# CLIを使う

- TTYを切り替える  
→Ctrl+Alt+F1~
- 端末エミュレータを使う  
→LXTerminal, Terminator, Konsoleなど

# CLIを使う (Continued)

- TTYを切り替える

VirtualBoxだとHostキー(初期設定では右Ctrl)+F1

```
Ubuntu 16.04.4 LTS lubuntu tty1
```

```
lubuntu login: _
```

ここでユーザ名とパスワードを入力するとログインしてCLIを使って操作できる

実はデスクトップ環境がないとこんな感じになる

Host+F7でGUI環境に戻る(tty7) (違うこともある)



# CLIを使う (Continued)

- TTYを切り替える  
一度TTYにログインすると明示的にログアウトするまではログインされたままなので注意

ex)

TTY1からGUI環境に戻ってもTTY1はログインされたままなので後で戻ってこれる  
デスクトップ環境で「ロック」してもほかのTTYにログインしていたら意味がない

- ログアウトするには

```
$ exit
```

# CLIを使う (Continued)

- 端末エミュレータを使う  
GUIからは普通これを使う

今回の仮想マシンに入っているのは"LXTerminal"

画面左下のメニュー>システムツール>LXTerminal  
をクリック

# 文字が小さい！

- 「編集」 > 「設定」 > 「端末のフォント」で変更できる
- あんまり見えずらい設定でやっていると目が死んでしまうので注意

# 使ってみる

```
$ echo こんにちはこんにちは  
こんにちはこんにちは
```

```
$ uname -a  
Linux lubuntu 4.13.0-36-generic #40~16.04.1-Ubuntu SMP  
Fri Feb 16 23:25:58 UTC 2018 x86_64 x86_64 x86_64  
GNU/Linux
```

# 使ってみる(Continued)

```
$ echo こんにちはこんにちは
```

- echo  
「コマンド」これはプログラムの名前だったり、シェルにもともとついていてコマンドだったりする(source, setなど)
- こんにちはこんにちは  
コマンドにあたえる値(引数ということもある)。

**echo**は与えられた文字列を表示するコマンドなのでこんな感じになる

# コマンドの強制終了

- Ctrl+Cでコマンドの実行をとめることができる

```
$ ping www.google.com
64 bytes from ...
64 bytes from ...
^C
--- www.google.com ping statistics ---
...
$
```

# 使ってみる(Continued)

- シェル？  
カーネルに対してユーザの操作を受ける「外殻」。CLIの有名なシェルに `bash` がある  
今使っているのがそれ
- コマンドに与える値  
空白区切り(改行はだめ)。  
単に値を書く以外にも `-a` や `--ignore-sth` のような形のものもある(スイッチ)  
`-a` の形のもののは合体させられることが多い(`-czf` みたいに)

もっと**CLI**



# ディレクトリ

- Windowsでは「フォルダ」と呼ばれる
- LinuxでもほかのOSとおなじで木構造のディレクトリ構造を持つ
- ファイルシステムの一番根っこの部分にあるディレクトリ(WinでいうE:\などディスクの直下)のことを「ルートディレクトリ」と呼ぶ

# 相対/絶対パス、カレントディレクトリ

- 絶対パス  
ルートディレクトリから見たファイル・ディレクトリの場所
- 相対パス  
カレントディレクトリから見たもの。前のディレクトリは `..` であらわす
- カレントディレクトリ  
作業するディレクトリのこと

# カレントディレクトリ

- `pwd`で確認(あるいはプロンプトを見て確認)

```
springcamp@lubuntu:~$ pwd  
/home/springcamp
```

`~`はホームディレクトリのこと(たいてい  
`/home/<ユーザ名>`だが違うものも使える)

# カレントディレクトリ (Continued)

- `cd`で変更

```
$ cd /etc  
$ pwd  
/etc
```

- ホームディレクトリに戻る

```
$ cd
```

# 環境変数

- OSあるいはセッション全体で共有されている変数

`$PATH`: コマンドを呼ぶときに探してくるパスの一覧

`$HOME`: ホームディレクトリへのパス

- 作る、代入することができる

```
$ export HOGE=foo
$ echo $HOGE
foo
```

シェルによってやり方が異なるので注意

# パスの中の空白

- コマンドに与えたい値に空白などが混じっているときは `"`、 `'` で囲む

```
$ cd "~/Documents/レポート 下書き/report_phy/report.tex"
```

この二つは実は意味が違うので注意(中に `"`、 `'` がある場合、中に環境変数を入れたい場合など)

# ワイルドカード

- 複数のファイルを指定したいときに便利

```
$ ls
memo.txt
memo1.txt
memo2.txt
memo2.txt.tmp
$ rm memo*.txt
$ ls
memo2.txt.tmp
```

- 「正規表現」というのがあがるが別物

# ワイルドカード(Continued)

- たまに想定していない挙動をすることがあるので  
注意  
「`memo.txt`残したかったのに」など
- `?`(任意の1文字)もある



# Linuxのユーザとグループ

- Linuxではユーザとグループで誰がどのファイルにアクセスできるか管理している
- ユーザID0はスーパーユーザーといい、原則何でもできる
- 複数のユーザをまとめて「グループ」に属させることができ、グループ全体で許可・禁止をできる
- スーパーユーザーになりたい場合は `sudo` を使う

```
$ sudo apt install build-essential  
[sudo] springcamp のパスワード:
```

# 所有者

あるファイルを「持っている」ユーザ

```
$ ls -l test.txt  
-rw-rw-r-- 1 springcamp adm 0 3月 8 15:50 test.txt
```

- 所有者は `springcamp`
- 所有グループは `adm`
- `chown` で変更できる

```
# chown ddns /etc/ddns.log
```

```
$ chown root:adm test.txt
```

# パーミッション

- あるファイルのアクセス許可情報(mode)ということもある
- 8進数
  - 512の位：ディレクトリかファイルかなど
  - 64の位：所有者ができること
  - 8の位：所有グループができること
  - 1の位：全員ができること

# パーミッション(Continued)

- 一桁だけ見ると
  - 0b100(4) : 読み込み(r)
  - 0b010(2) : 書き込み(w)
  - 0b001(1) : 実行、ディレクトリ探索(x)

これらのうち許可するものの論理和をとる  
たとえば書き込み+読み込みなら  
 $0b100 \text{ OR } 0b010 = 0b110 = 6$

# パーミッション(Continued)

- `chmod` で変えられる

```
$ chmod 755 /home/springcamp
```

自分はフルアクセス + ほかの人は読み込みと探索  
ディレクトリはxがないと中身がわからなくなってしまうので注意

```
$ chmod +x setup.sh
```

# パッケージマネージャ

- パッケージを管理するソフト  
→あるパッケージがあるパッケージをインストールしないと動かない(依存関係)などの問題を自動的に解決してくれたりもする
- たいていのディストリビューションが提供している  
→ `apt`、 `dnf`、 `pacman` など
- Debian系は `apt`

# apt

- リポジトリ情報の取得

```
$ sudo apt update
```

サーバ上のパッケージの情報を取得する  
定期的に行うと良い

- パッケージの検索

```
$ apt search python
```

キーワードを渡す

# apt (Continued)

- パッケージのインストール

```
$ sudo apt install build-essential
```

- パッケージの削除

```
$ sudo apt remove python
```

```
$ sudo apt purge openssh-server
```

**purge** は設定ファイルも削除する



# apt (Continued)

- 必要ないパッケージの削除

```
$ sudo apt autoremove
```

依存されなくなったパッケージを消す

- パッケージのアップグレード

```
$ sudo apt upgrade
```

`update` してからでないとも最新にならないので注意

# apt (Continued)

- パッケージのアップグレード(古いパッケージの削除を含む)

```
$ sudo apt dist-upgrade
```

Linuxのバージョンがあがった場合などに使う。  
UbuntuでGUIを使ってパッケージを更新するとこれが行われるらしい

# コマンド紹介

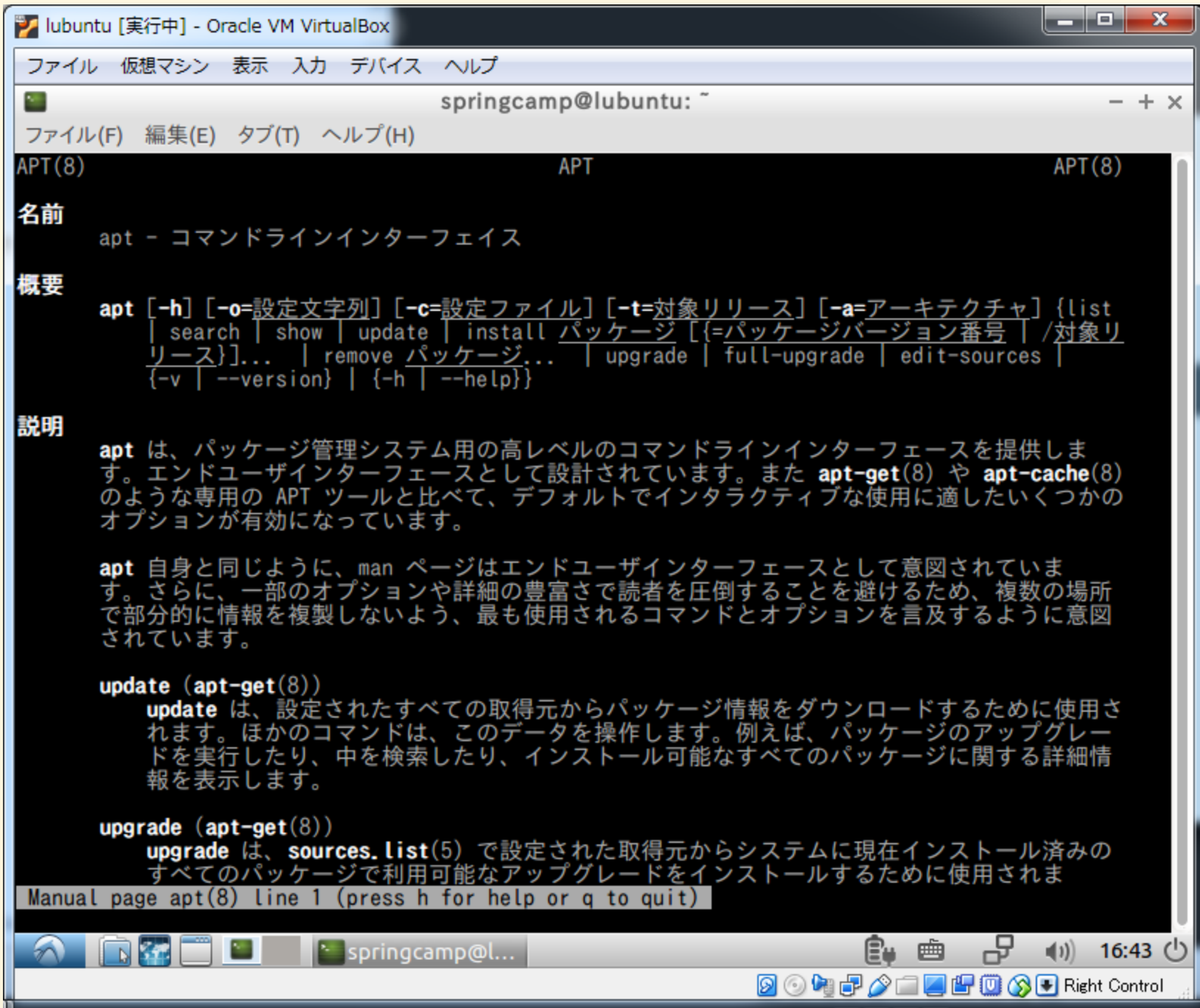
# ヘルプ

- `man` を使うとコマンドの使い方が表示される

```
$ man cp
```

- 英語だったりするので気合で読む
- `man` のヘルプは

```
$ man man
```



# ファイル操作

- ファイルのコピー

```
$ cp コピー元 コピー先
```

- ディレクトリ丸ごとコピー

```
$ cp -r コピー元 コピー先
```

- 移動

```
$ mv 移動元 移動先
```

# ファイル操作(Continued)

- シンボリックリンクを張る

```
$ ln -s リンク先 リンクの名前
```

- 削除(ゴミ箱には行かないので注意)

```
$ rm hoge.txt
```

- ディレクトリ丸ごと削除

```
$ rm -r ~/深淵へようこそ_pics
```

# ファイル操作(Continued)

- ディレクトリの作成

```
$ mkdir 深淵へようこそ_pics
```

- ディレクトリをパスも含めて作成

```
$ mkdir -p ~/Downloads/musics
```

Downloadsがなかったら作成される

- 空のディレクトリを削除

```
$ rmdir temp-dir
```



# CLI関連

- ディレクトリの内容を見る

```
$ ls /usr/bin
```

何もディレクトリを与えないとカレントディレクトリについて表示される

```
$ ls -lA
```

**-l**をつけるると詳細な情報を表示

**-A**をつけるると **..**, **.**以外の **.**から始まるファイル・ディレクトリも表示する

# CLI関連(Continued)

- カレントディレクトリの変更

```
cd /tmp/springcamp
```

- 画面の消去

```
$ clear
```

- ログアウト(エミュレータ終了)

```
$ exit
```

# 検索

- 標準入力またはファイルから検索する

```
$ ls /usr/bin | grep gcc
```

■ はコマンドの標準出力と入力をつないでくれる

# プロセス管理

- 実行中のプロセスをリスト化

```
$ ps -u ユーザ名
```

いろいろな呼び方があるので省略

- 実行中のプロセスを終了する

```
$ kill プロセスID
```

- 実行中のプロセスを名前で検索して終了

```
$ pkill firefox
```

# 圧縮

- ファイルをZip圧縮

```
$ zip hoge.zip ファイル ファイル .....
```

- ディレクトリ丸ごとZip圧縮

```
$ zip -r hoge.zip ディレクトリ
```

- ディレクトリをTAR+GunZipで圧縮

```
$ tar -czf hoge.tar.gz ディレクトリ
```

# 解凍

- Zipを解凍

```
$ unzip hoge.zip
```

- TARおよびTAR+任意のアーカイブを解凍

```
$ tar -xf hoge.tar.gz
```

# その他

- ファイルに標準出力を流し込む

```
$ echo hello > test.txt
```

- 標準入力にファイルを流し込む

```
$ foo < testcase.txt
```

- 両方やる

```
$ foo < testcase.txt > result.txt
```

# 超入門のその先



# アプリケーションの見つけ方

- (Ubuntuなら)システムツール>ソフトウェアセンターを使う
- ググる
- <https://wiki.archlinux.jp/index.php/アプリケーション一覧>
- 布教してもらう

下の二つが一番楽そうではある

# emacs と vim

- ソフトウェア開発で食べていきそうな人はどちらかを使ってみるのが個人的なお勧め
- もし何かの拍子で `vi` が起動してはまってしまったら  
`ESC` `:q!` `Enter` で幸せになれます

# 自分で環境を立ててみよう

- 忘れないうちに自分でLinux仮想マシンを構築してみると吉  
→自分の好きな環境にカスタマイズできるだけでなく、いろいろな知見を得ることもできる

# Linux実機入門

- WindowsとデュアルブートするとメジャーアップデートでLinuxが起動しなくなることがある  
→ブートローダーの再インストールで解決
- Windows用のデバイスは動かないこともある  
→インターネットで動くかどうかを事前に調べておく  
→USBメモリやキーボード、マウスはたいてい動く

# シングルボードコンピュータ

- 運が良いと数千円でLinuxマシンが手に入る
- Intel入ってないことが多いので動かないアプリがあるかもしれない

ex)

Raspberry Pi, Beagle Bone, Tinker Boardなどなど

# Reference

- Keyboard Shortcut for Changing Desktops in  
Lubuntu(Ask Ubuntu)  
<https://askubuntu.com/questions/821912/keyboard-shortcut-for-changing-desktops-in-lubuntu>
- Send Ctrl+Alt+F1, Ctrl+Alt+F7 into  
VirtualBox(Ask Ubuntu)  
<https://askubuntu.com/questions/542328/send-ctrlaltf1-ctrlaltf7-into-virtualbox>
- 環境変数(「分かりそう」で「分からない」でも  
「分かった」気になれるIT用語辞典)  
<http://wa3.i-3-i.info/word11027.html>

# Reference(Continued)

- Is using Software Updater the same thing as running apt-get update and apt-get upgrade?(Ask Ubuntu)

<https://askubuntu.com/questions/362872/is-using-software-updater-the-same-thing-as-running-apt-get-update-and-apt-get-u>

- Linux入門の入門(暗黒通信団、茗荷さくらほか)

# 参考になるサイト・書籍

- Ubuntu日本語フォーラム  
(<https://forums.ubuntulinux.jp/>)  
なんでも質問コーナー
- Arch Wiki(<https://wiki.archlinux.jp/>)  
Arch Linuxの公式ドキュメント。トラブル解決の参考になることがある
- 日経Linux(月刊誌)  
Linuxシステムの応用方法や使い方についての雑誌