

モバイル プログラミング2

本日の予定

午前

- Linux基礎
 - ユーザー権限
- Linuxコンソール
 - GUIとCUI
 - コマンド練習

午後

- ネットワーク基礎
 - 「プロトコル」と「ポート」
- HTTP演習
 - プロフィールページを作る

前回休んだ人

(°▽°)ｼ

まずは追いつこう

1. GitHubの資料見てね
2. 環境構築
3. 環境構築で困ったらすぐに聞いてください

PC借りた人

(°▽°)ｼ

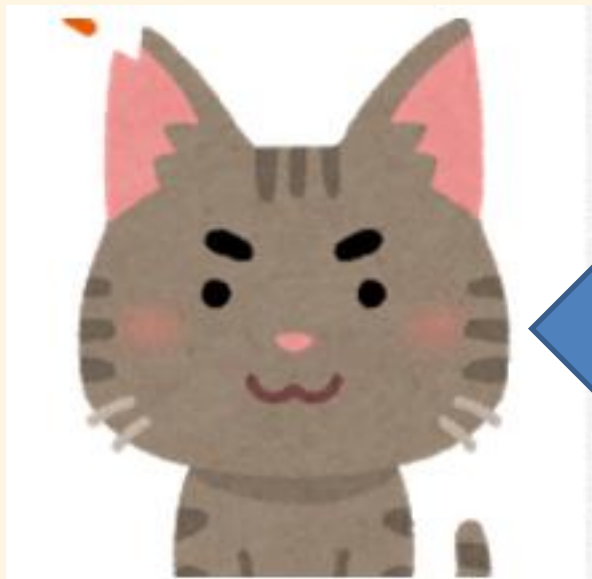
まずは追いつこう

1. 環境構築

2. 環境構築で困ったらすぐに聞いてください

前回の復習

AさんとBさんがデータを交換する
にはどうすれば良いだろう？



Aにゃん



Bにゃん

数百万人が一度にデータを交換するにはどうすれば良いだろう？



http://gigazine.net/news/20070129_lan/

サーバとは

- 24時間／365日
 - 稼働している
 - ネットワークに接続
- データを保存、配信
- たくさんの人が一度に利用

サーバ



<http://pc.watch.impress.co.jp/docs/2006/0907/dell.htm>
<http://www.dell.com/jp/business/p/poweredge-r410/pd>

データセンター



<http://www.bytez.nl/neem-ene-kijkje-in-het-datacenter-van-facebook/>

サーバの中身（ハード）



サーバの内部

- CPU
- メモリ
- ハードディスク(HDD)
- 電源ユニット
- ネットワークI/F

「パソコン」の中身（ソフト）

ブラウザや
ワード

その他
ソフト

OS (Windows, Mac)



「サーバ」の中身（ソフト）

サーバ用
ソフト

その他
ソフト

OS



サーバのOSとして使われる Linux



リーナス・トーバルズ氏

リーナスさんが作った
UNIXだから
『Linux』



Linux

- 無料
- オープンソース
- 世界中のエンジニアが開発に参加

Google



Linux基礎

その2



ユーザー権限

- Linuxは1台を複数人が使う前提
- 管理者と一般ユーザー



ユーザー権限

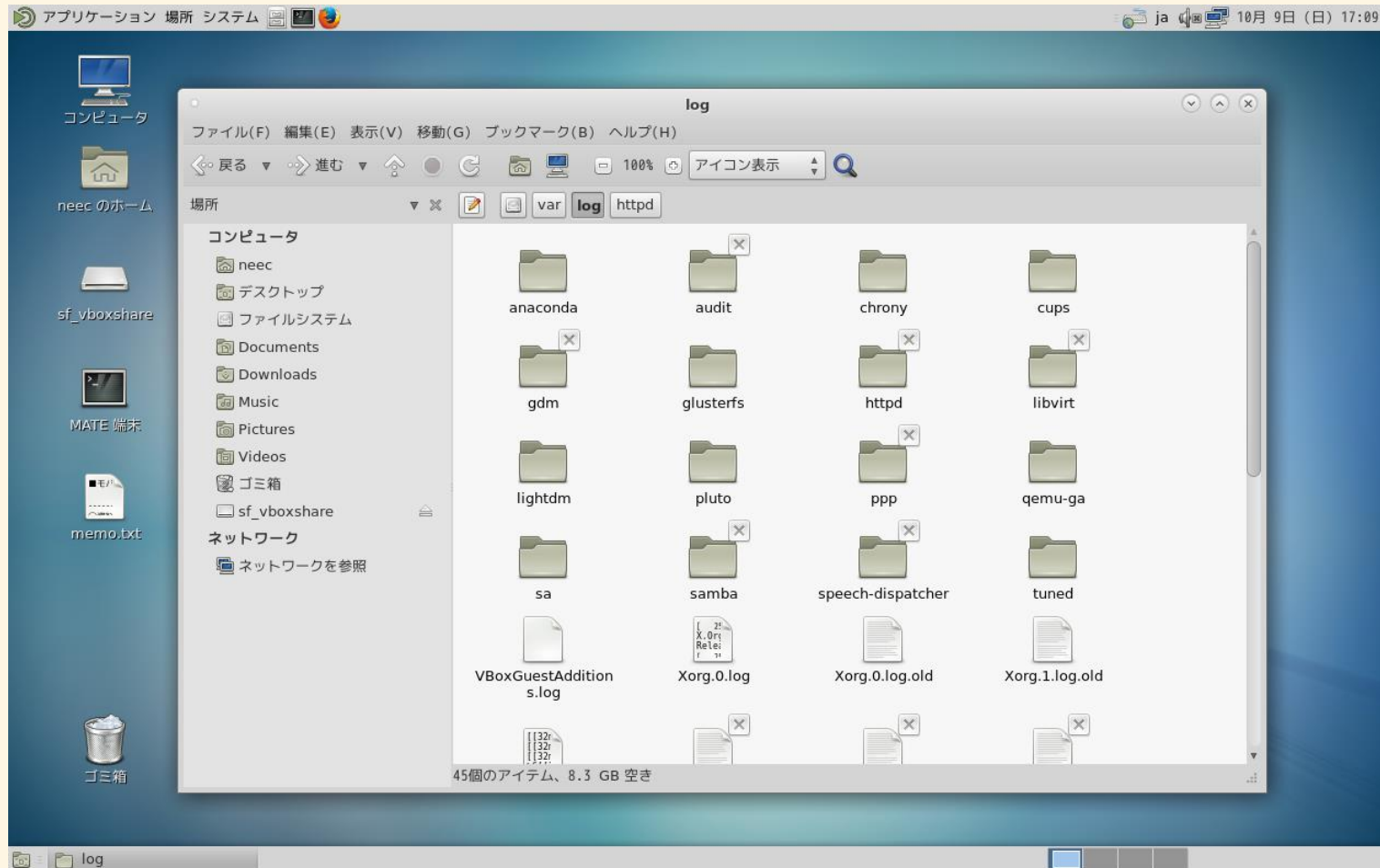


スーパーユーザー
(root)

一般ユーザー

やってみよう

/var/log/ の下のディレクトリ(フォルダ)やファイル



やってみよう

OSの設定変更



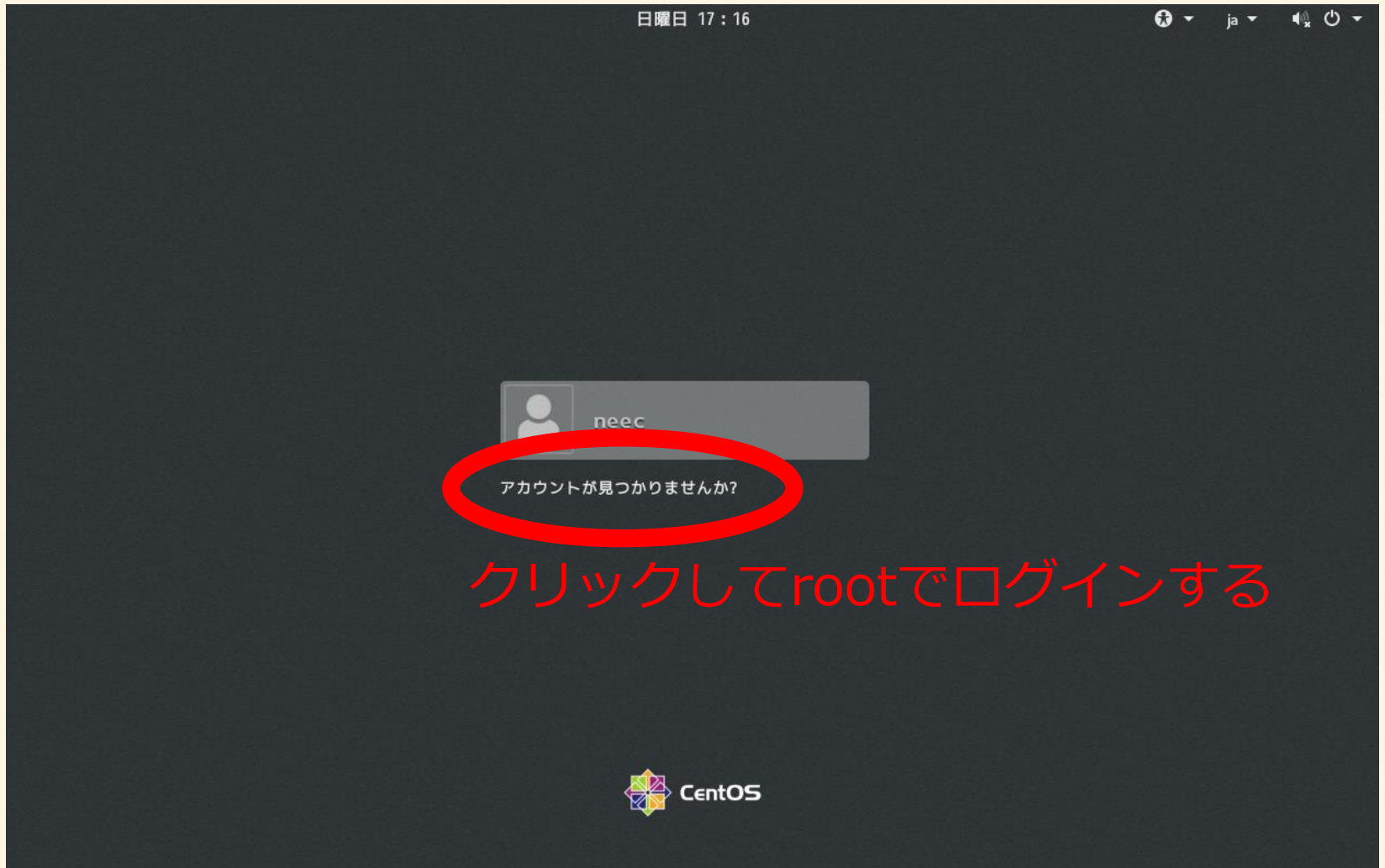
やってみよう



やってみよう



やってみよう



アカウントが見つかりませんか?

クリックしてrootでログインする



やってみよう

ユーザー名 : root

パスワード : root2016

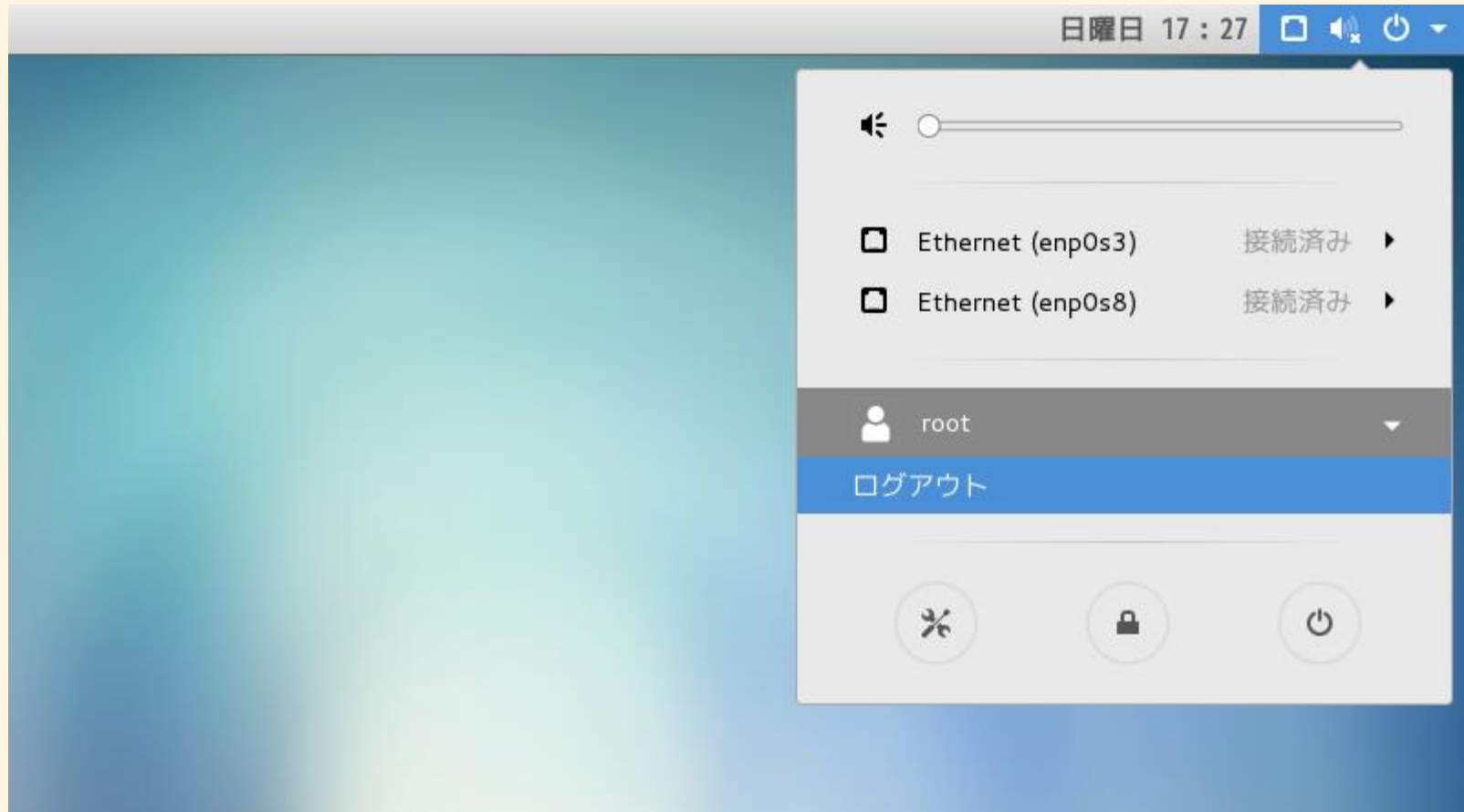


ユーザー名:

キャンセル 次へ

ログアウト

ログアウトし一般ユーザー(neec)に戻しておいてください



rootは特別な時だけ



スーパーユーザー
は何でも出来てし
まうので、**非常に
危険。**

普段は必ず一般
ユーザーで。

ユーザーの権限

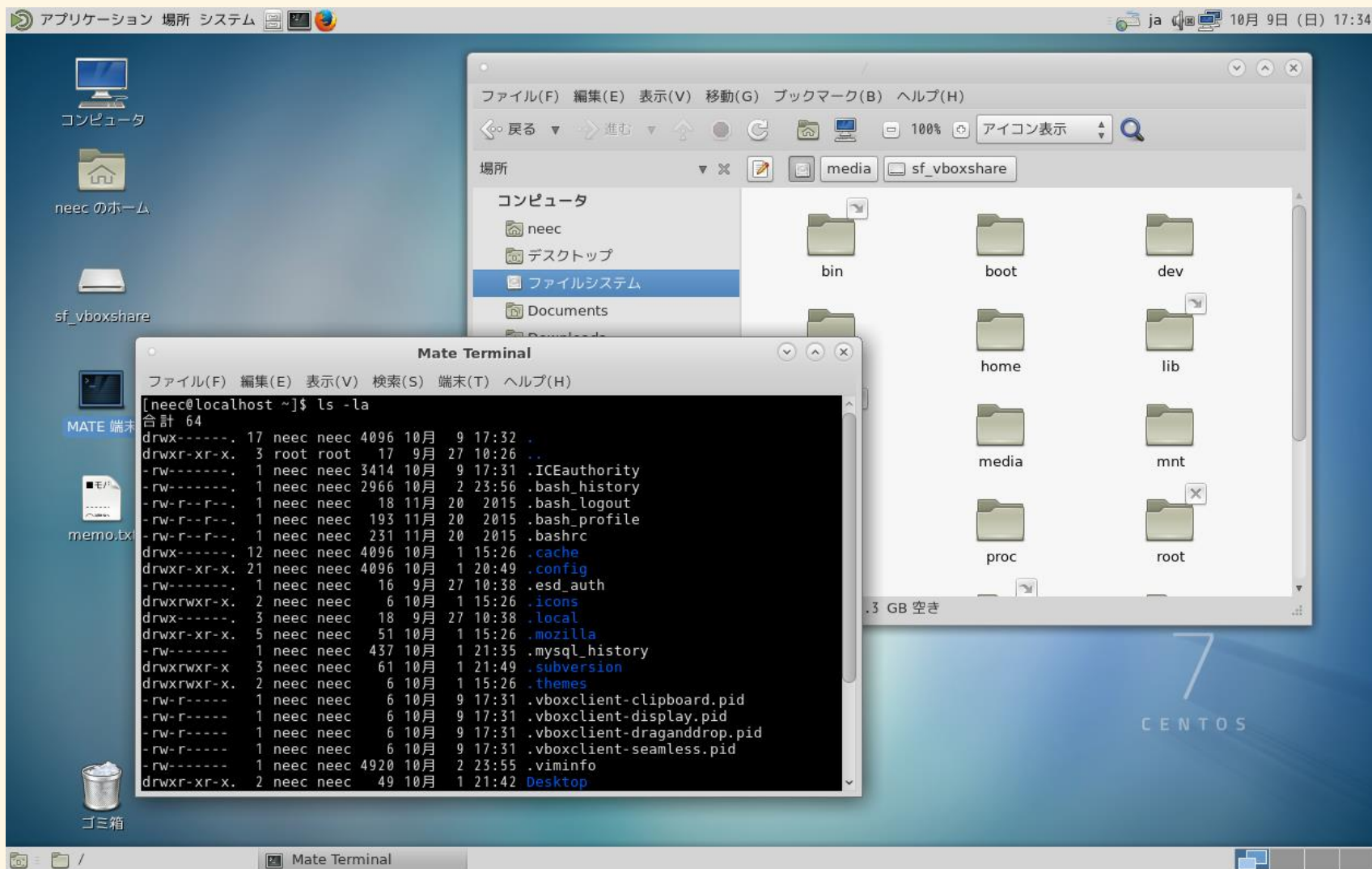
- 「スーパーユーザー」と「一般ユーザー」
- スーパーユーザー = root
- 普段の作業は一般ユーザー

Linuxコンソール



なぜCUI ?

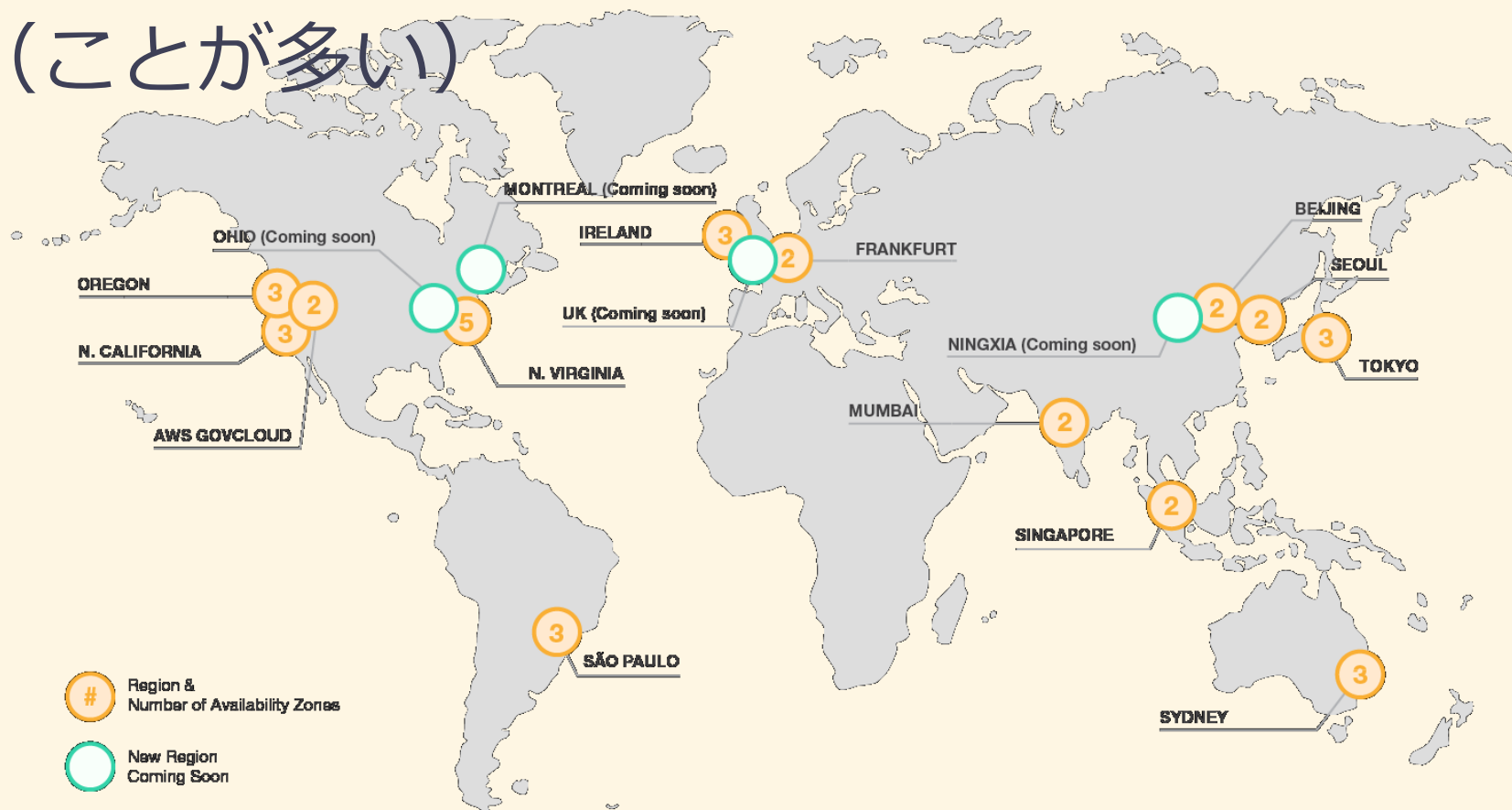
GUIとCUI



なぜCUIを使うの？

前提として、

サーバは遠く離れたところにある
(ことが多い)



<https://aws.amazon.com/jp/about-aws/global-infrastructure/>

なぜCUIを使うの？

PSの「リモートプレイ」方式なら？



<http://www.jp.playstation.com/support/torne/guide/psp/>

なぜCUIを使うの？

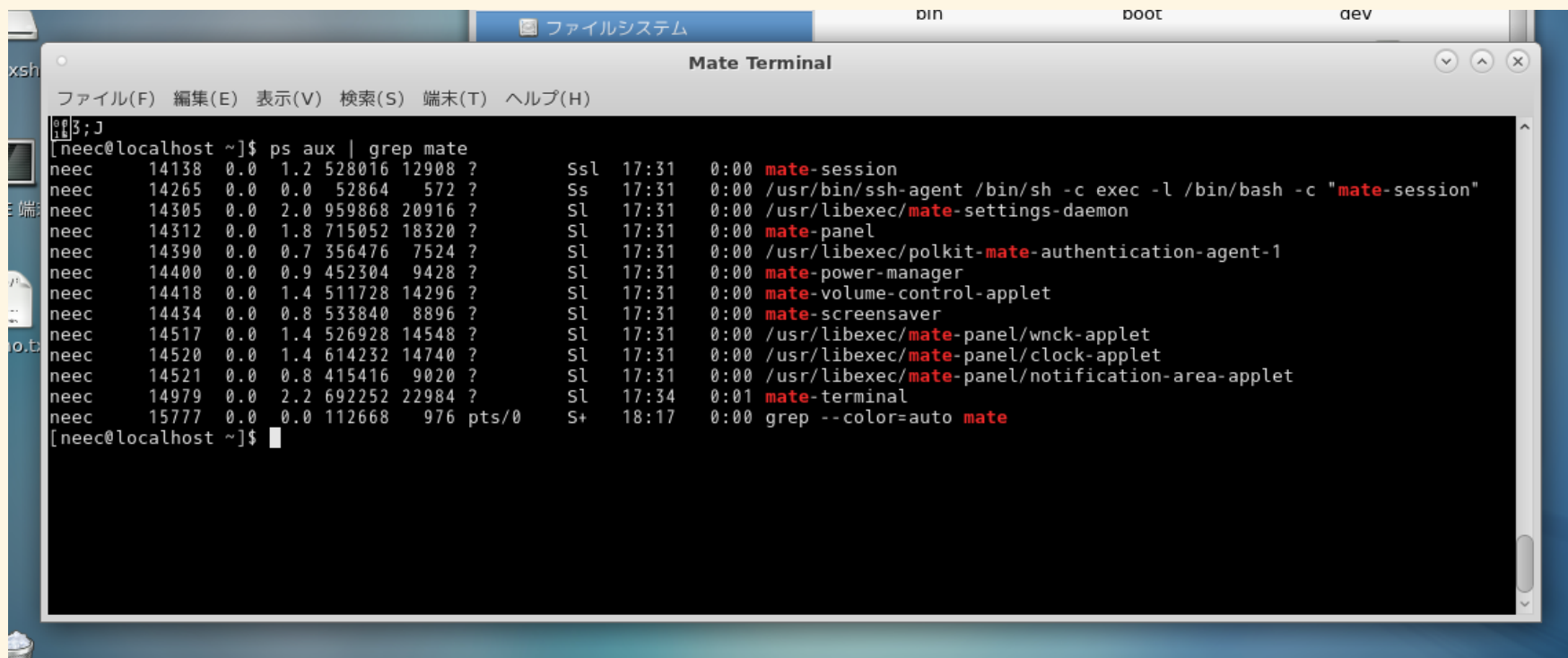
PCゲームのようなサーバクライアント方式なら？



http://jp.finalfantasyxiv.com/heavensward/#system__workshop

なぜCUIを使うの？

```
$ ps aux | grep mate
```



The screenshot shows a Mate Terminal window with the command `ps aux | grep mate` executed. The output lists various processes related to the Mate desktop environment, including the session, ssh-agent, settings-daemon, panel, authentication-agent, power-manager, volume-control-applet, screensaver, and terminal. The terminal window has a menu bar with options like File (F), Edit (E), View (V), Search (S), Terminal (T), and Help (H). The background shows a desktop environment with a file manager and other icons.

```
[neec@localhost ~]$ ps aux | grep mate
neec    14138  0.0  1.2 528016 12908 ?        Ssl  17:31   0:00  mate-session
neec    14265  0.0  0.0  52864   572 ?        Ss   17:31   0:00  /usr/bin/ssh-agent /bin/sh -c exec -l /bin/bash -c "mate-session"
neec    14305  0.0  2.0 959868 20916 ?        Sl   17:31   0:00  /usr/libexec/mate-settings-daemon
neec    14312  0.0  1.8 715052 18320 ?        Sl   17:31   0:00  mate-panel
neec    14390  0.0  0.7 356476  7524 ?        Sl   17:31   0:00  /usr/libexec/polkit-mate-authentication-agent-1
neec    14400  0.0  0.9 452304  9428 ?        Sl   17:31   0:00  mate-power-manager
neec    14418  0.0  1.4 511728 14296 ?        Sl   17:31   0:00  mate-volume-control-applet
neec    14434  0.0  0.8 533840  8896 ?        Sl   17:31   0:00  mate-screensaver
neec    14517  0.0  1.4 526928 14548 ?        Sl   17:31   0:00  /usr/libexec/mate-panel/wnck-applet
neec    14520  0.0  1.4 614232 14740 ?        Sl   17:31   0:00  /usr/libexec/mate-panel/clock-applet
neec    14521  0.0  0.8 415416  9020 ?        Sl   17:31   0:00  /usr/libexec/mate-panel/notification-area-applet
neec    14979  0.0  2.2 692252 22984 ?        Sl   17:34   0:01  mate-terminal
neec    15777  0.0  0.0 112668   976 pts/0    S+   18:17   0:00  grep --color=auto mate
[neec@localhost ~]$
```

なぜCUIを使うの？

```
$ ps aux | grep sshd
```



The screenshot shows a Mate Terminal window with a menu bar (File, Edit, View, Search, Terminal, Help) and a title bar (Mate Terminal). The terminal displays the command `ps aux | grep sshd` and its output, which lists two processes: `sshd` running as root and `grep` running as neec. The output is color-coded, with `sshd` in red and `grep` in red.

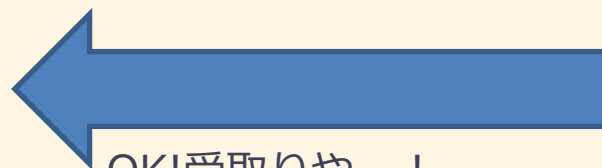
```
[neec@localhost ~]$ ps aux | grep sshd
root      1420   0.0   0.0  82560   988 ?        Ss   13:48   0:00 /usr/sbin/sshd -D
neec      15887  0.0   0.0 112664   976 pts/0    R+   18:21   0:00 grep --color=auto sshd
[neec@localhost ~]$
```


なぜCUIを使うの？

CUIなら文字情報だけ。
1つのコマンドで完結する。



ファイルの一覧ください



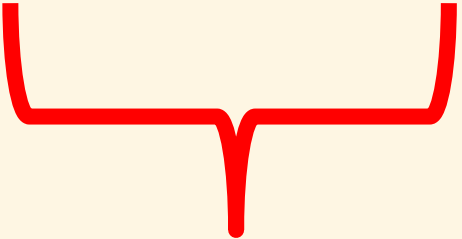
OK!受取りやー！



コマンド練習

書式

```
$ host twitter.com
```



コマンド



オプション

ファイル

ls

```
$ ls
```

- ファイルの一覧を表示

ls

```
$ ls Desktop
```

- 指定したディレクトリ内のファイル一覧を表示

ls

```
$ ls -a
```

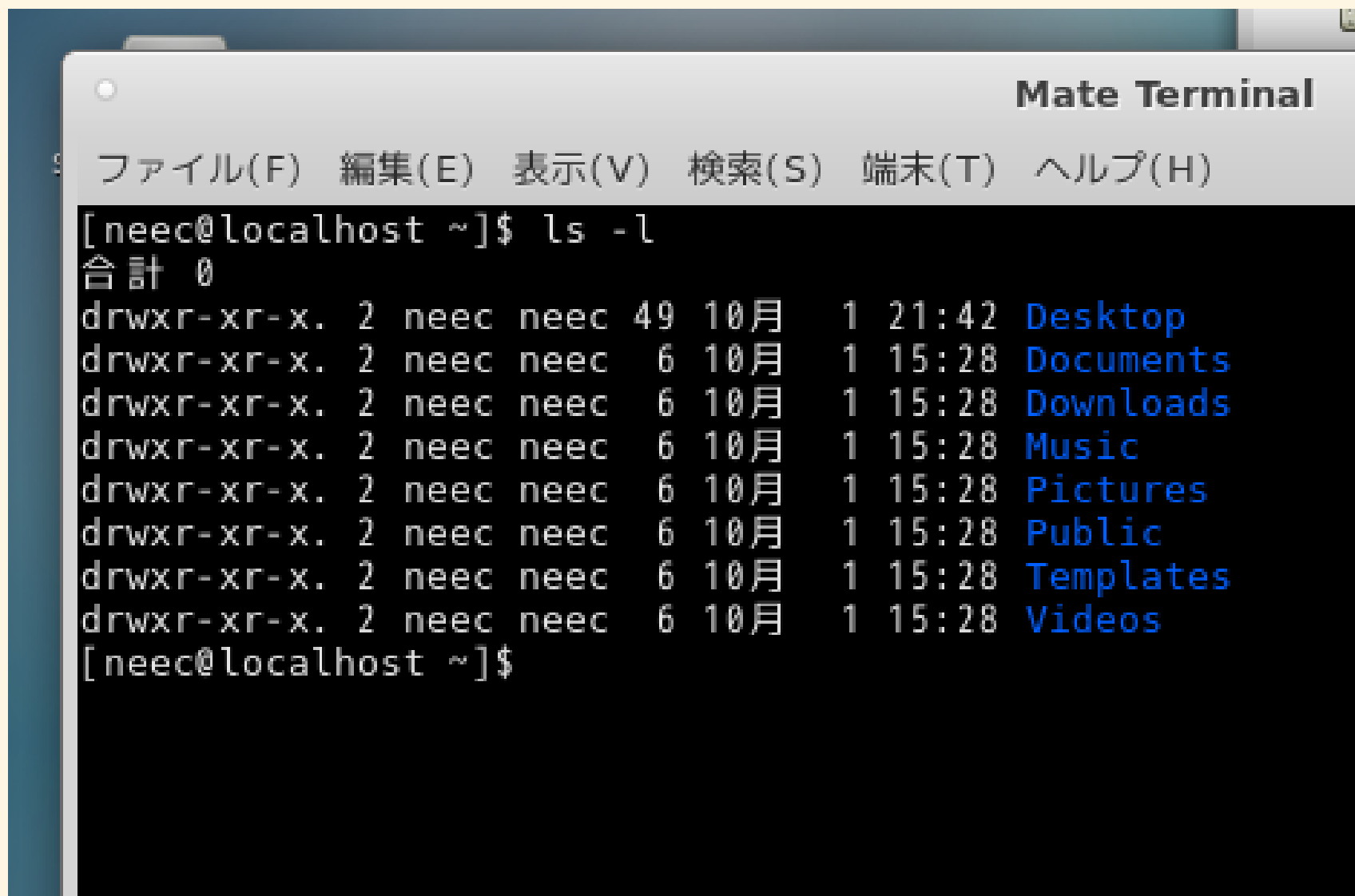
- -a をつけると隠しファイルも含めてすべてを表示

ls

```
$ ls -l
```

- -l をつけるとファイルの詳細情報も表示

ls



The image shows a screenshot of a terminal window titled "Mate Terminal". The window has a menu bar with the following items: ファイル(F), 編集(E), 表示(V), 検索(S), 端末(T), and ヘルプ(H). The terminal content shows the command `[neec@localhost ~]$ ls -l` being executed. The output lists the contents of the home directory with detailed permissions, owner, group, size, date, and filename. The files listed are Desktop, Documents, Downloads, Music, Pictures, Public, Templates, and Videos. The prompt `[neec@localhost ~]$` is shown at the bottom.

```
[neec@localhost ~]$ ls -l
合計 0
drwxr-xr-x. 2 neec neec 49 10月 1 21:42 Desktop
drwxr-xr-x. 2 neec neec  6 10月 1 15:28 Documents
drwxr-xr-x. 2 neec neec  6 10月 1 15:28 Downloads
drwxr-xr-x. 2 neec neec  6 10月 1 15:28 Music
drwxr-xr-x. 2 neec neec  6 10月 1 15:28 Pictures
drwxr-xr-x. 2 neec neec  6 10月 1 15:28 Public
drwxr-xr-x. 2 neec neec  6 10月 1 15:28 Templates
drwxr-xr-x. 2 neec neec  6 10月 1 15:28 Videos
[neec@localhost ~]$
```

ls

```
$ ls -l -a
```

```
$ ls -la
```

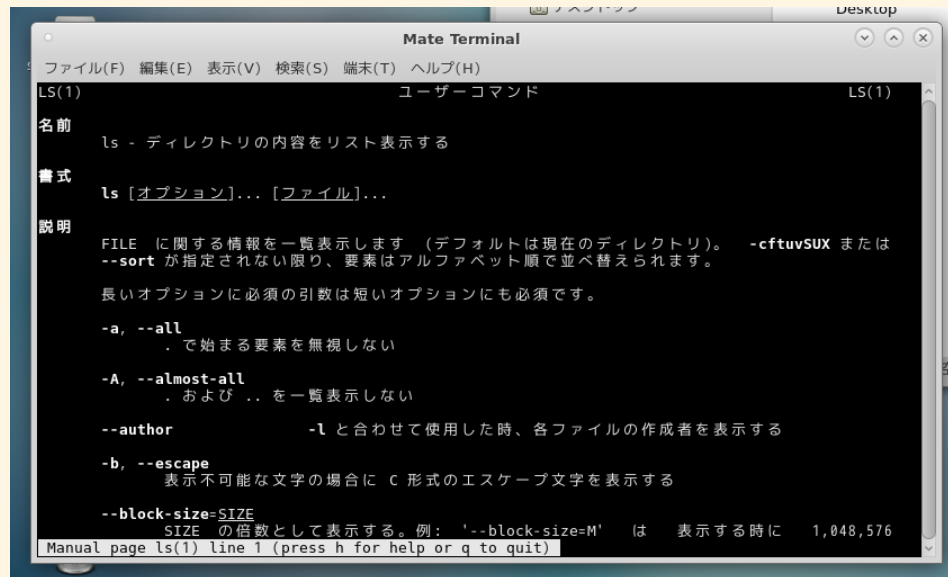
```
$ ls -la Desktop
```

- 同時に複数指定することも可能

man ls

\$ man ls

- 使い方を忘れたら man コマンド



```

Mate Terminal
ファイル(F) 編集(E) 表示(V) 検索(S) 端末(T) ヘルプ(H)
LS(1) ユーザーコマンド LS(1)

名前
ls - ディレクトリの内容をリスト表示する

書式
ls [オプション]... [ファイル]...

説明
FILE に関する情報を一覧表示します (デフォルトは現在のディレクトリ)。 -cftuvSUX または
--sort が指定されない限り、要素はアルファベット順で並べ替えられます。

長いオプションに必須の引数は短いオプションにも必須です。

-a, --all
    . で始まる要素を無視しない

-A, --almost-all
    . および .. を一覧表示しない

--author
    -l と合わせて使用した時、各ファイルの作成者を表示する

-b, --escape
    表示不可能な文字の場合に C 形式のエスケープ文字を表示する

--block-size=SIZE
    SIZE の倍数として表示する。例: '--block-size=M' は 表示する時に 1,048,576

Manual page ls(1) line 1 (press h for help or q to quit)
```

- スペースでページ送り
- ↑, ↓で行送り
- “q”で終了

cat

```
$ cat Desktop/memo.txt
```

- ファイルの内容を表示

ディレクトリ

cd

```
$ cd Desktop
```

- カレントディレクトリを移動

pwd

```
$ pwd
```

- カレントディレクトリのフルパスを表示

※カレントディレクトリは、正式にはカレントワーキングディレクトリ

※print working directory

ホームディレクトリ

```
$ cd ~
```

- ‘~’はホームディレクトリを表す特別な文字

ルートディレクトリ

```
$ cd /
```

- 最も上位に位置する階層をルートディレクトリ、または単にルートと呼びます

ディレクトリ

- 「カレント」
 - 今、自分がいる場所
- 「ホーム」
 - 最初の場所 (/home/neec)
- 「ルート」
 - 一番上

var → www → html

```
$ cd /var/www/html
```

- Linuxではディレクトリの階層構造を **'/'** で区切って記述

ひとつ上の階層へ行く

```
$ cd ..
```

- 一つ上の階層を '..' で表現する
- 自分自身は '.'

2つ上の階層へ行く

```
$ cd ../..
```

- '..'は'/'で区切っていくらでも指定できる

2つ上の階層にある
ディレクトリへ行く

```
$ cd /var/www/html
```

```
$ cd ../../log
```

```
$ pwd
```

絶対パスと相対パス

絶対パス

```
$ cd /var/www/html
```

相対パス

```
$ cd ../../log
```

その他
コマンド

mkdir

```
$ mkdir ディレクトリ名
```

- ディレクトリを作成します

cp

```
$ cp コピー元 コピー先
```

- ファイルをコピーします。
- -r をつけるとディレクトリも可能

rm

\$ rm ファイル

- ファイルを削除します。
- -r をつけるとディレクトリも可能

※remove direvtory

mv

\$ mv 移動元 移動先

- ファイル、ディレクトリを移動します。
- 応用するとリネーム

su と sudo

su

```
$ su ユーザー名
```

- 一時的にユーザーを変更します

su

```
$ su
```

- 一時的にrootになります
- 特別な事情がなければ“su -” と “-”オプションをつける

sudo

```
$ sudo 実行コマンド
```

- rootとしてコマンドを実行

Webサーバを起動する

```
$ sudo systemctl start httpd.service
```

- Webサーバソフトの「Apache」を起動します。
- ログイン時のパスワードを聞かれるので入力します。

※neec2016

Where is
Linux
Command?

/usr/bin

```
$ cd /usr/bin
```

```
$ ls
```

```
$ cd /usr/sbin
```

```
$ ls
```

環境変数 PATH

```
$ printenv
```

```
PATH=/usr/local/bin:/usr/local/sbin:  
/usr/bin:/usr/sbin:/bin:/sbin:/home/  
neec/.local/bin:/home/neec/bin
```

- 環境変数はOS内の「グローバル変数」のような物
- コマンドを実行 = プログラムを起動

UNIXという哲学

書籍「UNIXという考え方」



すべて

誕生

Amazonポイント: 85

マイストア ギフト券 タイムセール

JP 勝部麻季人さん アカウントサービス

本 詳細検索 ジャンル一覧 新刊・予約 Amazonランキング コミック・ラノベ 雑誌 文庫・新書 Amazon Student

本 > コンピュータ・IT > コンピュータサイエンス

 お客様は、2002/5/19にこの商品を注文しました。
[この注文を表示](#)

なか見!検索↓



UNIXという考え方—その設計思想と哲学

単行本
— 2001/2

Mike Gancarz (著), 芳尾 桂 (翻訳)

★★★★☆ 23件のカスタマーレビュー

▶ その他 () の形式およびエディションを表示する

単行本

¥ 1,728

¥ 1,299 より 14 中古品の出品
¥ 1,728 より 3 新品

10/10 月曜日 にお届けするには、今から**13 時間 41 分**以内に「お急ぎ便」または「当日お急ぎ便」を選択して注文を確定してください（有料オプション。Amazonプライム会員は無料）

定理1: スモール・イズ・ビューティフル

2.1 **定理1**: スモール・イズ・ビューティフル

プログラムを書くときは小さなものから始めて、それを小さなままに保っておく。簡単なフィルタプログラムでも、グラフィックスパッケージでも、巨大なデータベースを構築するときでも、同じく小さな実用的なプログラムにする。一つの巨大なプログラムにしようとする誘惑に負けないで、シンプルさを追及する。

伝統的なプログラマは、巨大なアメリカンプログラムを書きたいという欲求を内に秘めていることが多い。このようなプログラマがプロジェクトに携わると、数週間、数カ月、あるいは数年かけてでも、世界中のすべての問題を一つのプログラムで解決しようとする。コスト的にビジネスに見合わないだけでなく、何よりも彼らは現実を無視している。現実には、その場かぎりの小さな解決策でも解決できない問題はそれほど多くはない。ほとんどの場合、問題を完全に理解していないから巨大な解決策を実装しようとしてしまうのだ。

分かりやすい

小さなプログラムは分かりやすい

小さなプログラムの「目的」は、最小限のものに抑えられ、一つのことを上手くこなすことに焦点が合っている。2、3のアルゴリズムしか含まず、そのほとんどが目的の作業に直接関係している。

一方、巨大なプログラムは複雑になることが多く、その理解には困難が伴う。プログラムが大きくなればなるほど、そのプログラムは作者の手から遠ざかる。とても読めないような、ものすごいプログラムの行数が理解を超えてしまうのだ。例えば、作者でも、あるサブルーチンをどのファイルに置いたか分からなくなり、この変数をどこで相互参照しているかを忘れ、変数の目的が何だったかを忘れてしまう。このようなプログラムのデバッグは、悪夢以外のなにものでもない。

保守しやすい

小さなプログラムは保守しやすい

小さなプログラムは、分かりやすい。分かりやすいと保守も容易になる。プログラムを理解するのが保守の第一歩だ。このことは、ほとんどの人は聞いたことがあるだろう。しかし、保守の問題に無関心なプログラマは依然として多い。誰かが、書くためにかかったのと同じだけの時間を使って、やってくれるだろうと思っているのだ。

ほとんどのソフトウェア技術者は、他人のプログラムの保守で生活することには満足しない。以前のプログラムの修正よりも、新しいプログラムを書くことのほうがよい仕事だと考えている。おそらく、その考えは正しいだろう。だが、不幸なことに、ユーザーは同じようには見てはくれない。ユーザーは、最初からプログラムがきちんと動くと思っている。動かなかったら大変だ。その会社はユーザーの信頼を失うことになる。うまく保守できない会社は長く生き残ることはできない。

システムリソースにやさしい

小さなプログラムはシステムリソースにやさしい

小さなプログラムの実行イメージはわずかなメモリしか占有せず、オペレーティングシステムもメモリを割り当てやすくなる。よって、スワップやページング*といった作業が劇的に減り、性能が大幅に向上する。こういうときにUNIX世界では、そのプログラムは「軽い」と表現する。つまり、小さなプログラムは軽いプロセスとみなされる。

大きなプログラムは実行イメージも大きく、実行するためにロードしたとき、オペレーティングシステムにかなり負荷がかかる。ページングやスワップが避けられず、性能にも悪影響が出る。オペレーティングシステムの設計者は、大きなプログラムからのリソース要求に配慮して、ダイナミックローディングや共有ライブラリといった仕組みを用意して対処する。しかし、それは対症療法であって、問題そのものを解決することにはならない。

どこからが「巨大」？

- 関数呼び出しに渡されるパラメータの数が、画面の一行では収まらなくなる。
- サブルーチンコードが一画面を超える。または、**8.5インチ×11インチの標準紙一枚を超える。フォントを小さくし、大型のワークステーション用モニタの大きなウィンドウに表示すれば、制限を少しは緩和できるが、やりすぎは禁物だ。
- プログラム中のコメントを読まないと、サブルーチンが何をやっているのか思い出せない。
- lsコマンドを実行すると、画面がスクロールしてモジュール名が端から消えていってしまう（lsは、他のオペレーティングシステムでいうDIRに相当するUNIXコマンドだ）。
- プログラム中のグローバル変数を定義するために、あるファイルが不格好にも膨れ上がってしまった。
- まだプログラムを開発中なのに、表示されたエラーメッセージがどのような条件で出力されるようにしたのか覚えていない。
- 整理するためにソースコードを印刷している自分に気が付いた。