

LinuC レベル1 技術解説無料セミナー

～LinuC レベル1 受験に向けての準備とポイント解説～

2019/09/01

河原木忠司（かわらぎただし）



■河原木忠司（かわらぎただし）

- 20数年ほど、講師/エンジニアとして活動しております。
- 最近は講師、研修コンサルティング、執筆業に従事させていただいております。
- サーバーインフラ系のコース、セキュリティ系のコースを中心に担当させていただいております。
- 好きなもの
 - パンダ
 - 妻
 - うずらの卵
 - 音楽





書籍のご案内

- 2019年7月、久々に書籍を出版させていただきました。
「最短突破 LinuCレベル1 合格教本」 (技術評論社)
 - 本会場では割引で販売させていただいておりますので、よろしければお手にとってご覧ください。
 - 本セミナーで利用するデモ環境は、本書籍の付録データを利用しております。





1. LinuC Level1試験について

- ◆試験概要と特徴
- ◆主な試験範囲等
- ◆学習環境の準備

2. ポイント解説

- ◆ポイント解説の前に・・・
- ◆シェル環境の基本的な使い方と活用（主題103/主題106）
-----※間に15分間の休憩を挟みます-----
- ◆ユーザー管理とパーミッション（主題104/主題107）
- ◆セキュリティ管理業務の実施（主題110）



1. LinuC Level1試験について



LinuC Level1について

- 詳しくは公式サイトにて・・・
<https://linuc.org/linuc/levels.html>
- レベル1～3があり、レベル1はLinuxについての基本操作/システム管理についての技術を問う内容
- LPI-Japanで問題を管理しているなので、正しい日本語での出題が期待できる。



LinuCレベル1

LinuC-1

Linuxシステムの構築・運用・管理

実務に必要なLinuxの基本操作とシステム管理が行えるエンジニアであることを証明できます。

該当試験

101試験 / 102試験

※LinuCレベル1に認定されるためには101試験と102試験の両方に合格する必要があります。

勉強期間目安

1か月～3か月程度

(※初めての方の場合の目安)

他業種で初心者から始めた取得者も多数！



Linux環境について

■様々なシステム環境で利用されている

■サーバー



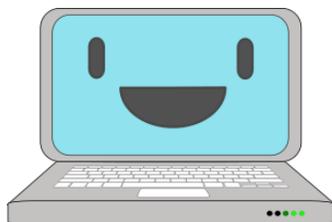
■モバイル 端末



■IoT端末



■ユーザー PC



■通信機器



Linuxについての知識を身につけると様々な場
面で、活かすことができる。



LinuC試験について

- 90分/約60問のCBT試験
- PearsonVueが配信
 - 全国各地のテストセンターで受験可能
 - <https://www.pearsonvue.co.jp/linuc>
 - ペーパーテストで受験できるイベントもあり。
 - <https://lpi.or.jp/pbt/20190928/>
- 選択問題（択一/複数選択）のほか、記入問題もあり。
 - うろ覚えだと正解を導き出しづらい
 - 終わった問題も再確認可能
- レベル1は101試験/102試験の2つに合格すると認定

一斉
ペーパーテスト
スタート

LinuC / OSS-DB / HTML5 / OPCEL
公開ペーパー試験のご案内

試験日	2019年9月28日(土)
会場	日本電子専門学校 (東京都新宿区)

Linux環境を実際に触りながら学習を進めるのがおすすめ！



101試験

- 主題101 : システムアーキテクチャ
- 主題102 : Linuxのインストールとパッケージ管理
- 主題103 : GNUとUnixのコマンド
- 主題104 : デバイス、Linuxファイルシステム、ファイルシステム階層標準

システムの起動管理、パッケージ管理、基本コマンド、ファイルシステム管理などを出題

102試験

- 主題105 : シェル、スクリプト、およびデータ管理
- 主題106 : ユーザインターフェイスとデスクトップ
- 主題107 : 管理業務
- 主題108 : 重要なシステムサービス
- 主題109 : ネットワークの基礎
- 主題110 : セキュリティ

シェルの活用法やGUI環境についての基本の他、システム管理の基本手法ならびにそれに付随する内容を出題



■仮想化環境を用意するのがおすすめ！

- 実機上で、仮想マシンを起動し、Linux環境を構築
- 複数のLinux環境を同一マシン上で利用することができる。
- 複数の仮想マシンを同時に起動し、ネットワーク接続を利用した構成の検証も可能

■仮想化環境を利用するために必要なソフトウェア・機能

- Hyper-V (Windows)
- VMWare Workstation Player (Windows/Linux)
- **Oracle VirtualBox (Windows/macOS/Linux/Solaris)**

主題102ではRed Hat系（CentOSなど）とDebian系（Debian Linuxなど）のパッケージ管理方式がそれぞれ出題

主題109ネットワークの基本知識のほか、主題110セキュリティなどでもネットワーク接続を前提とした内容が出題

本セミナーでは、Oracle VirtualBoxを利用した仮想化環境を利用して、デモ操作をご紹介します。



仮想化環境の構築に必要な手順

1. 仮想化環境のインストール

<https://www.oracle.com/jp/virtualization/virtualbox/>

2. Linuxディストリビューションのインストール用isoファイルの入手

<https://www.centos.org/download/>

3. 仮想マシンの作成

4. Linuxディストリビューションのインストール

■ Linuxディストリビューション

Linuxの配布形態。Linuxカーネルを「OS」として動作させるために必要な様々なパッケージを追加して配布。

Red Hat系、Debian系といった系統がある。今回のセミナーでは、Red Hat系の代表的なディストリビューションであるCentOSを利用してデモを実施





仮想化環境の構築に必要な手順

1. 仮想化環境のインストール

<https://www.oracle.com/jp/virtualization/virtualbox/>

2. Linuxディストリビューションのインストール用isoファイルの入手

<https://www.centos.org/download/>

3. 仮想マシンの作成

4. Linuxディストリビューションのインストール

Oracle VM VirtualBox マネージャー

ファイル(E) 仮想マシン(M) ヘルプ(H)

ツール

環境設定(P) インポート エクスポート **新規(N)** 追加(A)

ようこそVirtualBoxへ!

このアプリケーション上で、仮想マシンの作成を行います。仮想マシンの作成には、このアプリケーションの「新規」ボタンをクリックする必要があります。

① [新規]をクリックし、仮想マシンの作成画面を表示

仮想マシンの作成

名前とオペレーティングシステム

新しい仮想マシンの記述名と保存フォルダーを指定し、インストールするオペレーティングシステムのタイプを選択してください。入力した名前はVirtualBoxでこのマシンを特定するのに使われます。

名前: CentOS6

マシンフォルダー: C:\Users\wh#VirtualBox

タイプ(T): Linux

バージョン(V): Red Hat (64-bit)

② 仮想マシンの名前や作成するディスクサイズ等を指定し、仮想マシンを作成



仮想化環境の構築に必要な手順

1. 仮想化環境のインストール

<https://www.oracle.com/jp/virtualization/virtualbox/>

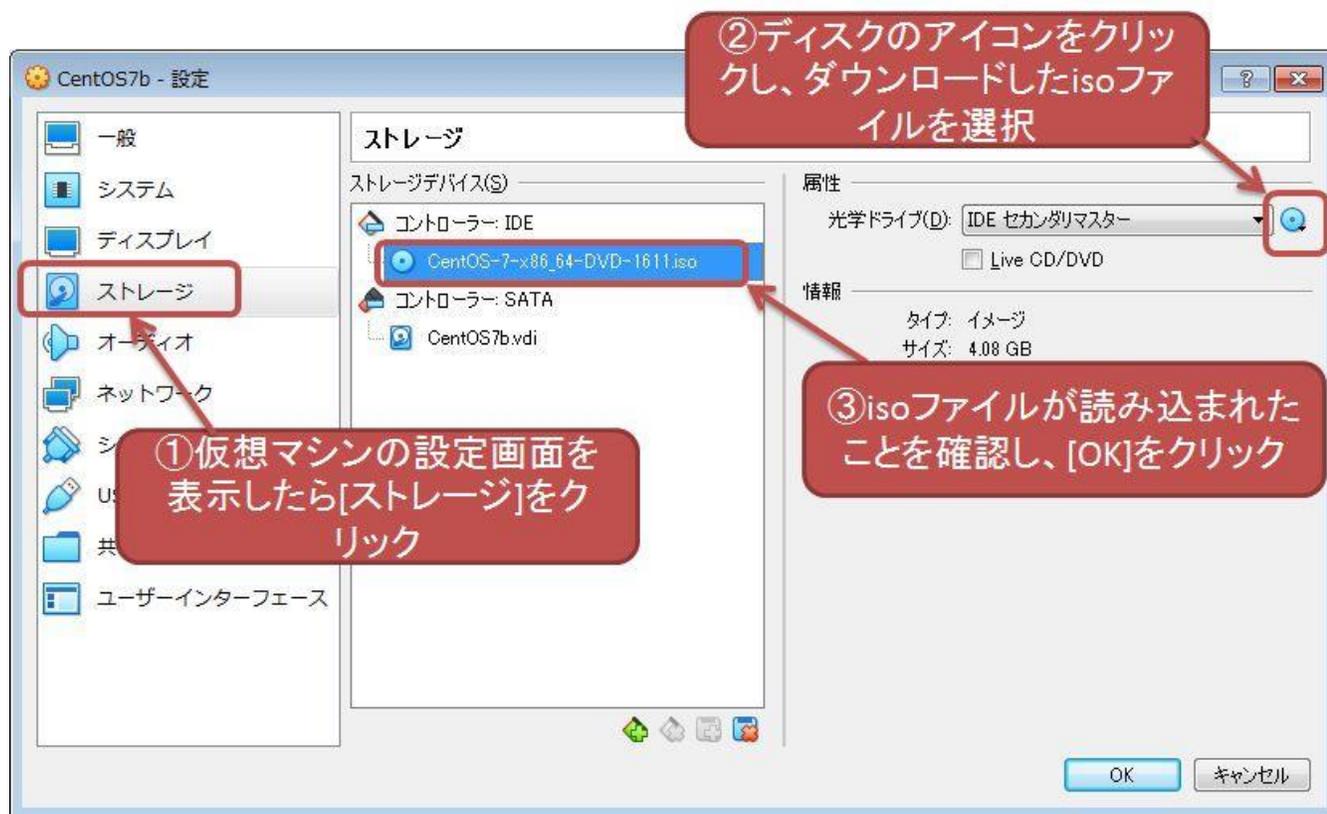
2. Linuxディストリビューションのインストール用isoファイルの入手

<https://www.centos.org/download/>

3. 仮想マシンの作成

4. Linuxディストリビューションのインストール

拙書「最短突破 LinuCレベル1 合格教本」(以降、「合格教本」)の付録には、Oracle VirtualBox上で動作する仮想マシンデータを用意しておりますので、2~4の手順は不要です。





仮想化環境の構築に必要な手順

1. 仮想化環境のインストール

<https://www.oracle.com/jp/virtualization/virtualbox/>

2. Linuxディストリビューションのインストール用isoファイルの入手

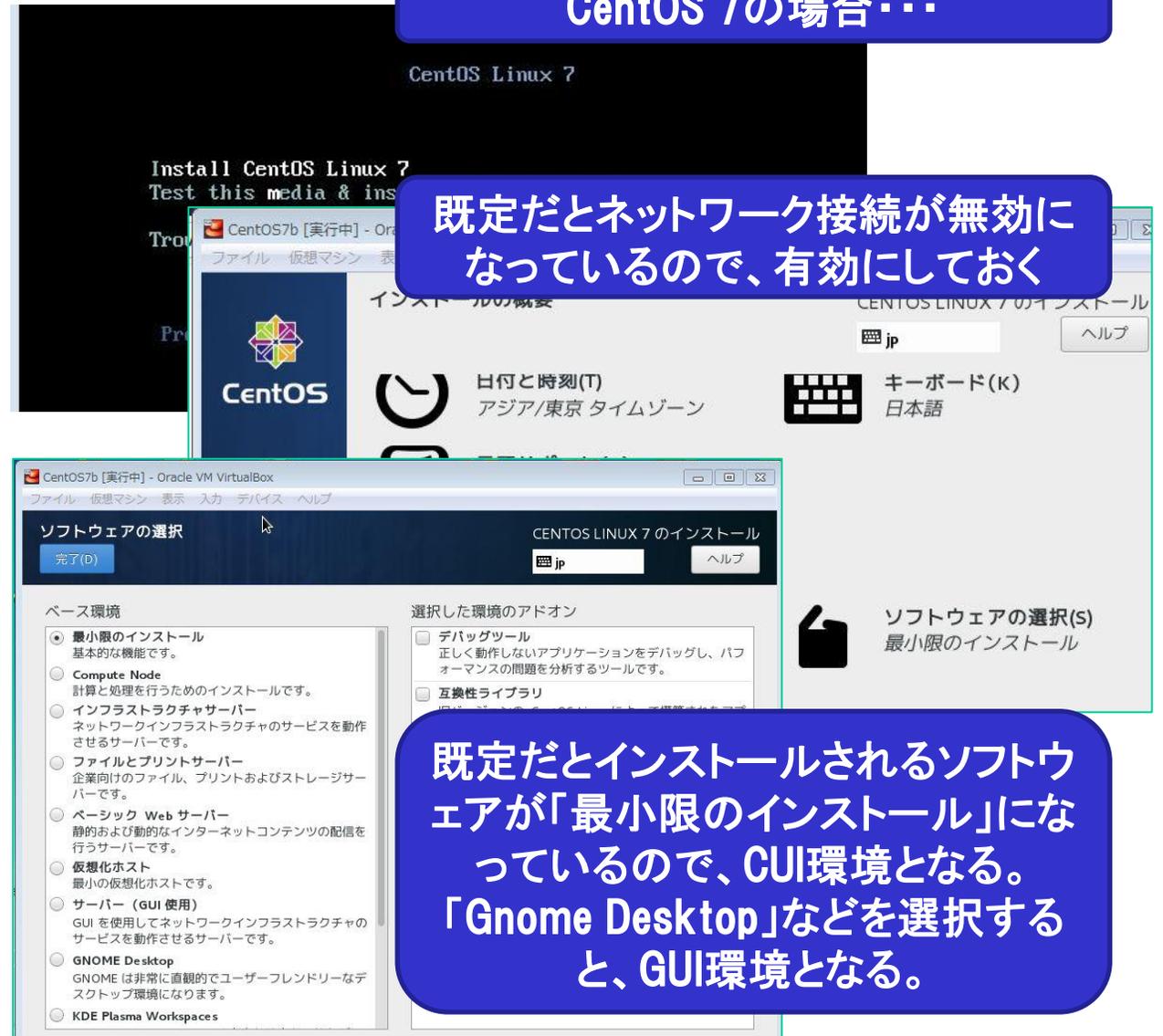
<https://www.centos.org/download/>

3. 仮想マシンの作成

4. Linuxディストリビューションのインストール

拙書「最短突破 LinuCレベル1 合格教本」(以降、「合格教本」)の付録には、Oracle VirtualBox上で動作する仮想マシンデータを用意しておりますので、2~4の手順は不要です。

CentOS 7の場合...



既定だとネットワーク接続が無効になっているので、有効にしておく

既定だとインストールされるソフトウェアが「最小限のインストール」になっているので、GUI環境となる。「Gnome Desktop」などを選択すると、GUI環境となる。



仮想化環境の利用

```

ragi — root@centos6:~ — ssh root@192.168.56.10
[ApplenoMacBook-Air-4:~ ragi$ ssh root@192.168.56.10
[root@192.168.56.10's password:
Last login: Wed Jul 10 00:50:59 2019
[root@centos6 ~]#

```

sshコマンドを利用して接続



GUI環境で実行

仮想マシンを直接操作しても、ターミナル経由で接続して操作しても構わないが、学習環境としては日本語を表示できる環境を推奨。

```

192.168.140.129:22 - root@localhost:~ VT
ファイル(E) 編集(E) 設定(S) コントロール(O) ウィンドウ(W) ヘルプ(H)
Last login: Tue Oct 12 09:45:57 2010
[root@localhost ~]# ls badname.txt
ls: badname.txt: そのようなファイルやディレクトリはありません
[root@localhost ~]#

```

teratermなどのターミナルソフトを利用して接続

実機

ssh

```

CentOS 55 - VMware Player ファイル(E) 仮想マシン(V) ヘルプ(H)
[root@localhost ~]# ls badname.txt
ls: badname.txt: No such file or directory
[root@localhost ~]#

```

CUI環境で実行

仮想マシン

このほか、AWSやGCPといったクラウド上のサービスを利用して、サーバーを構成して、利用するという方法もあり。



2. ポイント解説

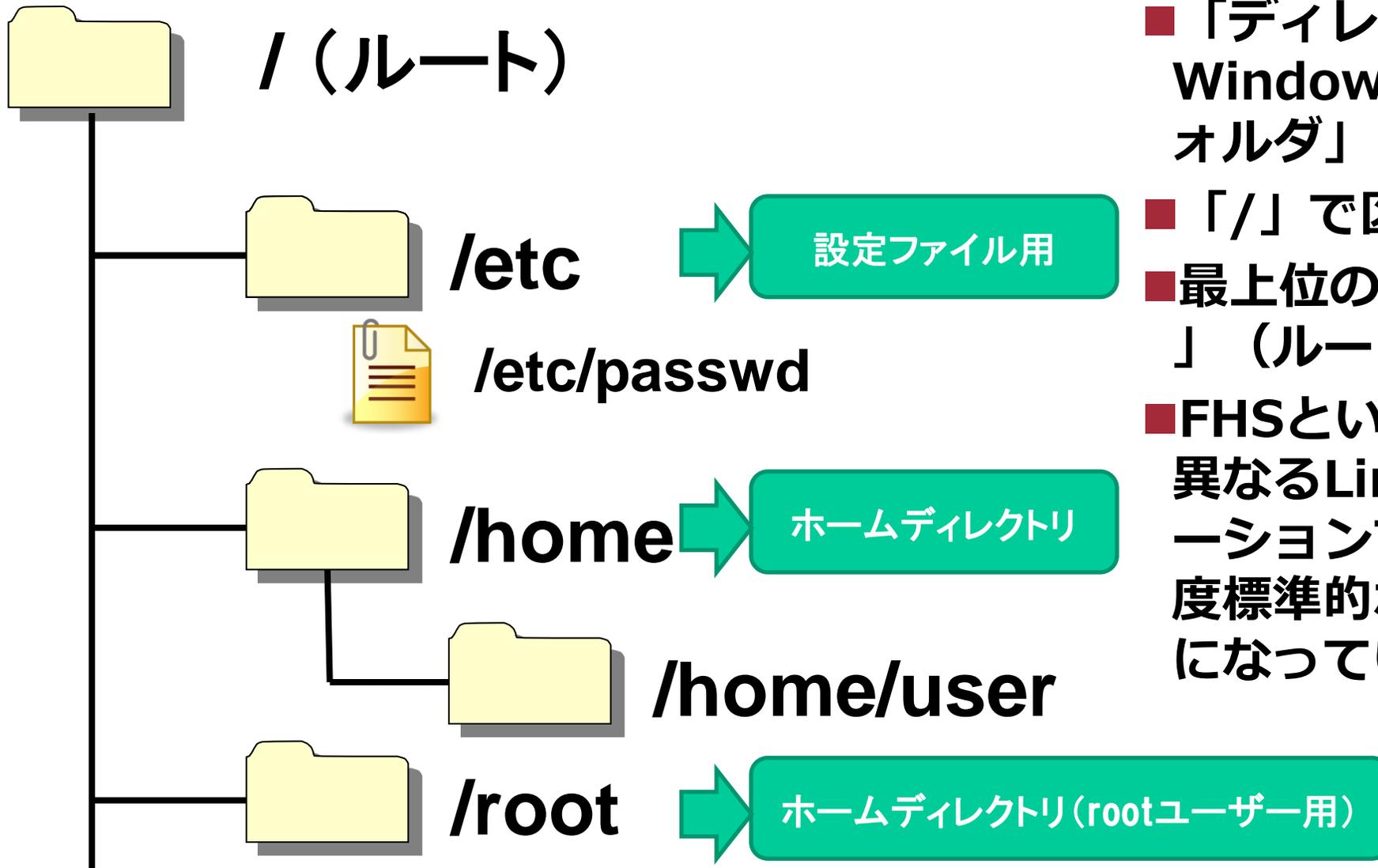


■Linuxに関する基本知識の確認

- ディレクトリ構造
- 基本コマンド
- rootユーザーについて



ディレクトリ構造



- 「ディレクトリ」 = WindowsやmacOSでの「フォルダ」
- 「/」で区切って表記
- 最上位のディレクトリは「/」（ルートディレクトリ）
- FHSという標準規格があり、異なるLinuxディストリビューションであっても、ある程度標準的なディレクトリ構造になっている。



基本コマンド

コマンドと引数	意味
cat ファイル名	テキストファイルの内容を表示
ls ディレクトリ/ファイル名	ディレクトリの内容を表示 -l : 指定したファイル/ディレクトリの詳細情報を表示
mkdir ディレクトリ名	ディレクトリの作成
head ファイル名	テキストファイルの先頭10行を表示 -n 数値 : 指定した行数表示
tail ファイル名	テキストファイルの末尾10行を表示 -n 数値 : 指定した行数表示
grep キーワード ファイル名	テキストファイルの中から指定したキーワードが書かれた行だけを表示 -v : 指定したキーワードが <u>書かれていない</u> 行を表示
echo 文字列	指定した文字列の表示。シェルスクリプト内で指定してメッセージを表示したり、リダイレクトと組み合わせて、ファイルの生成に利用
wc -l ファイル名	ファイルの行数を表示

Linuxでは設定情報を設定ファイルに記述して、システム管理を行うため、テキストファイルを操作するコマンドをしっかりと使いこなす必要がある。



rootユーザーについて

- UNIX/Linux系システムにおけるシステム管理用ユーザー
- システムにおけるすべての操作が可能
- パーMISSIONの制約がなく、すべてのファイルやディレクトリに対して、読み書きが可能

```
[root@centos6 ~]# ls -l /etc/shadow
-----. 1 root root 1194 5月 20 23:27
2019 /etc/shadow
[root@centos6 ~]# head -3 /etc/shadow
root:$6$qowcBD7fJyfaCzar$WYO8K8FXzcmc
6T9z3wwSwL.4a96XBjpZB3ZSe2g6MASPf06O
VFf5eh8z1arjD2fZA3t82KDojTX04piJ5kjL71:1
7854:0:99999:7:::
bin:*:17246:0:99999:7:::
daemon:*:17246:0:99999:7:::
```

すべてのユーザーが権限がないと
なっている/etc/shadow ファイル
に対して、読み書きが可能



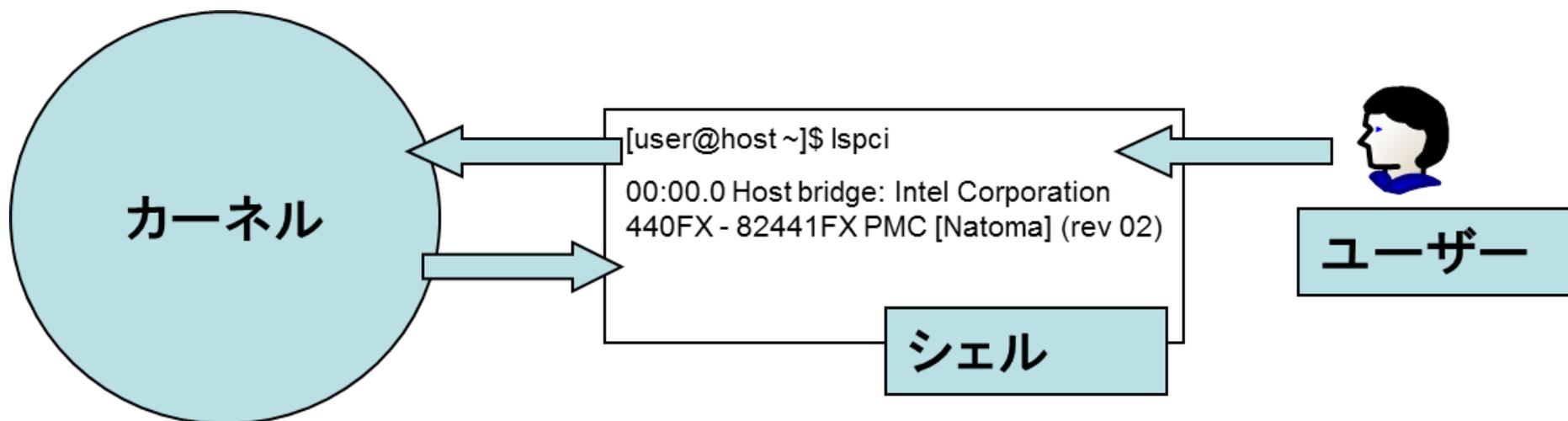
■ 「シェル」

- ユーザーがOS（カーネル）に命令をする手段を提供
- コマンドシェル：「コマンドを実行して、命令をする」環境を提供

■ Linuxで利用できる主なシェル

- **bash**
- dash
- csh

CentOSをはじめ、多くのLinuxディストリビューションで標準のログインシェルとして利用
LinuC試験ではbash環境を前提とした問題が出題





- /etc/passwdファイル（ユーザー情報を登録している設定ファイル）の末尾がログインシェル。
- ログイン時（GUI環境の場合、端末起動時）にここで指定されているシェルが起動し、ユーザーのコマンド入力を受け付ける。
- bashは高機能なシェルで、様々なコマンド入力支援機能、スクリプト処理機能を持つ。

```
[root@centos6 ~]# grep sh$ /etc/passwd  
root:x:0:0:root:/root:/bin/bash  
testuser:x:500:500::/home/testuser:/bin/bash  
postgres:x:26:26:PostgreSQL Server:/var/lib/pgsql:/bin/bash  
dashuser:x:501:501::/home/dashuser:/bin/dash
```

「合格教本」のCentOS6環境には、root、testuser、dashuserと③ユーザーが登録されています。

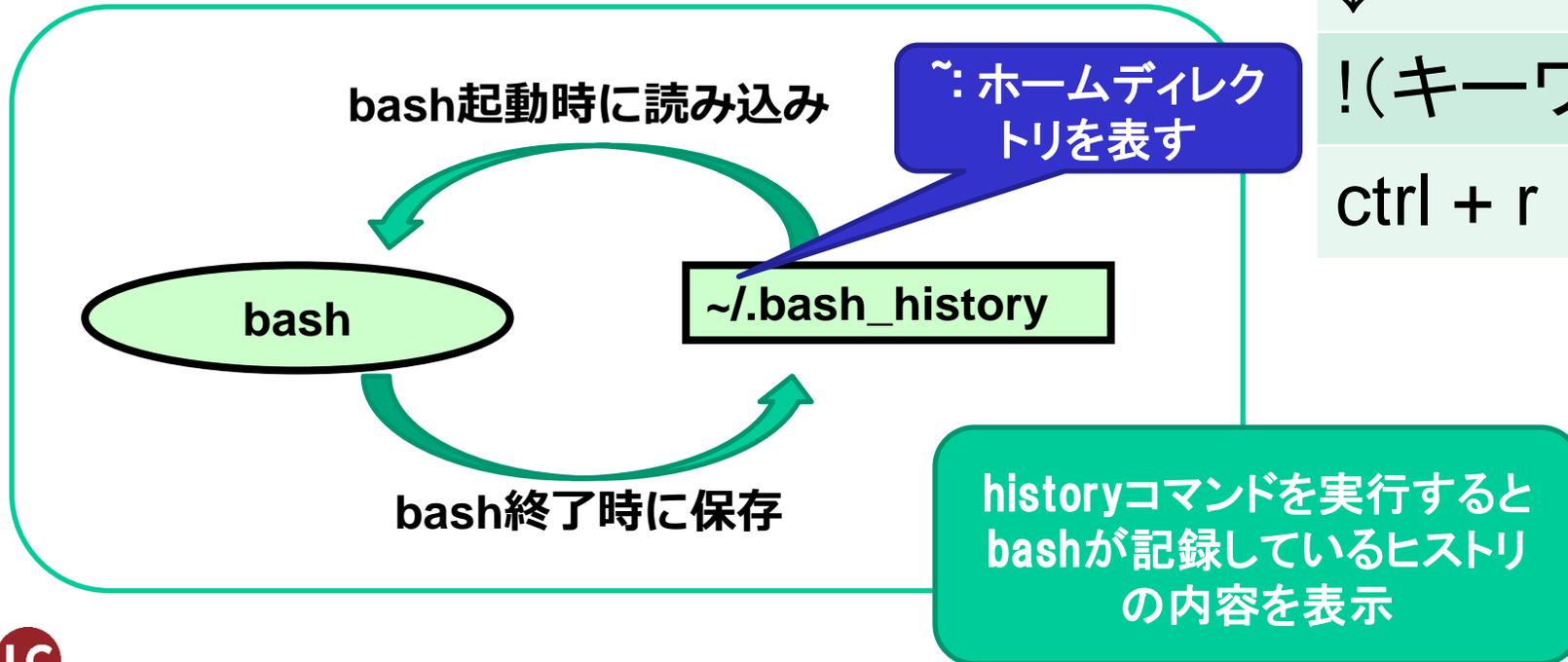


■ ヒストリ機能

- 入力されたコマンドを再現して、利用
- 起動中の履歴はシェルが記憶し、シェル終了時にファイルに保存

■ 主な操作

キー操作	意味
↑	コマンド履歴を再現
↓	コマンド履歴を戻す
!(キーワード)	コマンド履歴から検索
ctrl + r	後方検索





■後方検索が便利！

- コマンドの先頭部分だけではなく、引数なども検索可能

ctrl + rを押すと、専用のプロンプトが表示されるので、検索したいキーワードを入力

入力したキーワードに応じて、コマンドを再現

```
(reverse-i-search) `shadow': ls -l /etc/shadow  
[root@centos6 ~]# ls -l /etc/shadow  
-----. 1 root root 1194  5月 20 23:27 2019 /etc/shadow
```

- コマンドが再現されたら・・・

キー操作	意味
enter	コマンドを実行
ctrl + r	前を検索
esc	コマンドを実行せず、元のシェルに戻る ※ctrl + cを押すと、コマンドを消去



■メタキャラクタの抑止

- 前提となるファイル : /etc/httpd/conf/httpd.conf
- Apache httpdの設定ファイル
- 今回のデモ環境のものは1000行ほどの設定ファイルとなっている。

```
[root@centos6 ~]# cat /etc/httpd/conf/httpd.conf
```

```
#  
# This is the main Apache server configuration file. It contains the  
# configuration directives that give the server its instructions.  
:  
# Do NOT add a slash at the end of the directory path.  
#  
ServerRoot "/etc/httpd"  
:
```

「#」がついている行
はコメント行



■メタキャラクターの抑止

- httpd.confのコメント行以外を表示する構文例

-v: 否定条件

「#」をそのまま指定すると、右側がコメントとして扱われてしまうので、効果を抑止(無効化)

```
[root@centos6 ~]# grep -v '#' /etc/httpd/conf/httpd.conf
```

```
ServerTokens OS
```

```
ServerRoot "/etc/httpd"
```

```
:
```

◆ 囲まれた部分のメタキャラクターを抑止

- ‘~’ (シングルクォーテーション)
- “~” (ダブルクォーテーション)

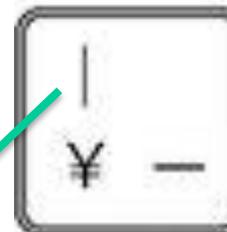
◆ 次の文字のメタキャラクターを抑止

- \~ (バックスラッシュ)



■パイプの利用

パイプの前(左)に指定したコマンドの出力結果(標準出力)を次のコマンドの標準入力に送る



```
[root@centos6 ~]# grep usb /var/log/messages | tail -3
```

```
Aug 24 13:31:13 centos6 kernel: usbcore: registered new interface driver hiddev  
Aug 24 13:31:13 centos6 kernel: usbcore: registered new interface driver usbhid  
Aug 24 13:31:13 centos6 kernel: usbhid: v2.6:USB HID core driver
```

/var/log/messagesはRedHat系の環境で起動時のメッセージやその他全般的な情報が格納されるログ
起動時に「usb」というキーワードがつく行が多く出力されるので、その中から末尾3行を出力している構文例。



■メタキャラクタの抑止

- httpd.confのコメント行と空白行以外を表示する構文例

^: 行頭、\$: 行末を表す
「^\$」= 行頭と行末だけ存在する行 = 空白行

```
[root@centos6 ~]# grep -v '#' /etc/httpd/conf/httpd.conf | grep -v '^$'  
ServerTokens OS  
ServerRoot "/etc/httpd"  
PidFile run/httpd.pid  
Timeout 60  
KeepAlive Off  
:
```

grepで、複数の条件両方に合致する行(AND条件)だけ表示するには、この例のようにパイプを活用する必要がある。
※OR条件であれば、-eオプションで指定



■ シェルスクリプト

- テキストファイル内に書かれたコマンドを実行
- 定期的に繰り返す処理を記述し、cronでスケジューリングして実行
- ファイルに実行権限を付与して、実行

```
[root@centos6 ~]# vi backup.sh
#!/bin/bash
tar cvf /backup/home-`date +%m%d`.tar /home
[root@centos6 ~]# chmod a+x backup.sh
[root@centos6 ~]# ./backup.sh
tar: メンバ名から先頭の `/' を取り除きます
/home/
/home/testuser/
:
[root@centos6 ~]# ls /backup/
home-0825.tar
```

/homeの内容を「home-日付.tar」というファイルにアーカイブ(バックアップ)する

実行権限を付与
※chmodコマンドは後ほど説明します

カレントディレクトリにあるスクリプトを実行
※ファイル名を指定するとスクリプトを実行



■以下のコマンドでユーザーの登録・削除といった管理が可能

- **useradd** [オプション] ユーザー名
 - ユーザーの登録
- **usermod** [オプション] ユーザー名
 - ユーザー設定の変更
- **userdel** [オプション] ユーザー名
 - ユーザーの削除
 - -rでホームディレクトリも一緒に削除
- **passwd** [オプション] [ユーザー名]
 - パスワードの指定
 - ユーザー名を省略するとログインしているユーザー自身のパスワードを変更
 - この操作に限り、一般ユーザーでも実行可能
 - パスワードが設定されていないユーザーはログイン不可



オプション	説明
-u UID	UIDを設定
-g グループ名/GID	プライマリグループを設定
-c コメント	コメントを登録
-d ディレクトリ	ホームディレクトリを設定
-s シェル	ログインシェルを設定



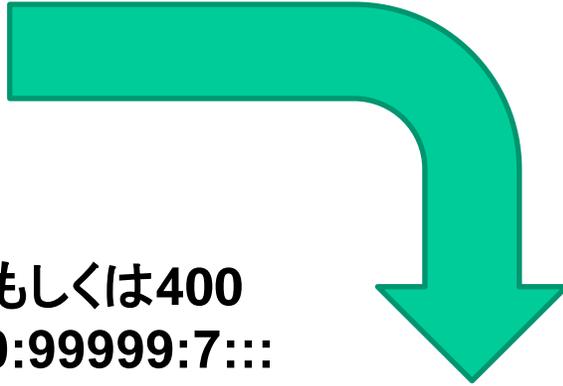
ユーザー管理とパーミッション（主題104/主題107）

■登録したユーザーの情報は次の設定ファイルに格納

- **/etc/passwd** : ユーザー情報を保存。パーミッションは644
test01:x:500:600::/home/test01:/bin/bash

ユーザー名

- **/etc/shadow** : パスワード情報を保存。パーミッションは000もしくは400
test01:\$6\$HTUBwjdV\$w7YdIKDg/3KRz73D2...:14952:0:99999:7:::



/etc/passwd
の記述内容

列	項目	説明
1	ユーザー	ユーザー名
2	パスワード	パスワード。passwdファイルでは「x」と記述されており、/etc/shadowで管理
3	UID	ユーザーに割り当てられているID
4	GID	ユーザーがプライマリグループとして所属しているグループのID
5	コメント	コメント。ユーザーの本名などを記載
6	ホームディレクトリ	ユーザーのホームディレクトリ
7	ログインシェル	ログインしたときに起動するシェル



■以下のスクリプトを作成し、実行するとtestuser1~testuser99というユーザーをまとめて作成

```
[root@centos6 ~]# cat useradd.sh
```

```
#!/bin/bash
```

```
for num in `seq 99`
```

1~99という連番を生成し、処理を繰り返す

```
do
```

```
  useradd testuser${num}
```

\${num}の部分に連番を格納

```
  echo password | passwd --stdin testuser${num}
```

```
done
```

```
[root@centos6 ~]# chmod a+x useradd.sh
```

```
[root@centos6 ~]# ./useradd.sh
```

標準入力(stdin)に送られた文字列をパスワードとして設定

ユーザー testuser1 のパスワードを変更。

passwd: 全ての認証トークンが正しく更新できました。

ユーザー testuser2 のパスワードを変更。

passwd: 全ての認証トークンが正しく更新できました。

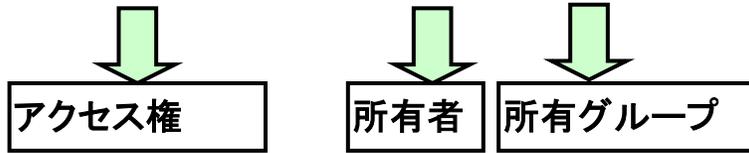
:



ユーザー管理とパーミッション (主題104/主題107)

```
[root@localhost ~]# ls -l sample.txt
```

```
-rw-r----- 1 root users 26 12月 9 22:52 sample.txt
```



640

r	4	読み取り
w	2	書き込み
x	1	実行

- $6 = 4 + 2$ (読み取り・書き込みが可能)

■ chmodコマンドの例

- `chmod 640 sample.txt`
- `chmod o+r sample.txt`



■それぞれの権限の具体例

	ファイル	ディレクトリ
r	ファイル内容の読み取り 例> cat sample.txt	ディレクトリ内のファイル一覧の読み取り 例> ls /sampledir
w	ファイル内容の編集 例> vi sample.txt	ディレクトリ内にファイルを作成・削除 例> rm /sampledir/sample.txt
x	ファイルの実行 例> /bin/ls	ディレクトリにアクセス可能 例> cd /sampledir ディレクトリ内の属性情報の読み取り 例> ls -l /sampledir



■以下のファイルは一般ユーザーでは読み取りできない

```
[root@centos6 ~]# mkdir -m 600 /sampledir
```

-m: パーミッションを指定

```
[root@centos6 ~]# echo test > /sampledir/test.txt
```

```
[root@centos6 ~]# ls -ld /sampledir
```

```
drw----- 2 root root 4096  8月 24 14:54 2019 /sampledir
```

```
[root@centos6 ~]# ls -l /sampledir/test.txt
```

```
-rw-r--r-- 1 root root 5  8月 24 14:54 2019 /sampledir/test.txt
```

```
[testuser@centos6 ~]$ cat /sampledir/test.txt
```

```
cat: /sampledir/test.txt: 許可がありません
```



セキュリティ管理業務の実施（主題110）

■rootユーザーの利用

- 実運用環境では、rootユーザーでのログインは最小限にする
 - 誤操作が発生したときのダメージ
 - セキュリティ上の理由により、rootユーザーのパスワード情報を知っているユーザーを最小限にする
- そのための設定: rootでのログインを無効化
 - ディストリビューションによっては、既定でrootでのログインはできない
 - sshdの設定 (/etc/ssh/sshd_config)
 - PermitRootLogin no
 - 一般ユーザーでログインし、必要なときにroot権限を利用
 - suコマンドで切り替え

```
[testuser@centos6 ~]$ su -
```

パスワード: 

```
[root@centos6 ~]#
```

su - [ユーザー名]

■ 指定したユーザー環境に切り替え

- ユーザー名を省略すると、rootユーザーの環境に切り替え
- rootユーザーからほかのユーザーに切り替えるときに限り、パスワード入力不要



■sudoコマンドの利用

- 管理者権限で実行したいコマンド構文の前に指定して利用
- rootユーザーのパスワードの入力は不要
- 事前に許可されたユーザーが、許可された操作だけ実行できる
 - visudoコマンドで実行可能なユーザー/コマンドを指定
 - vi(エディタ)が起動し、/etc/sudoersファイルを編集
 - 許可するユーザー/コマンドを追記する
 - 保存終了時にエラーチェック
- sudoの設定を有効にしたら・・・
 - suコマンドを実行できるユーザーを制限する（pamの設定: Lv2範囲）



■sudo利用のための設定例

```
[root@centos6 ~]# visudo
```

```
testuser    ALL=(ALL)    /sbin/shutdown
```

許可するユーザー名 実行可能なホスト=(実行ユーザー) 許可するコマンド

```
[testuser@centos6 ~]$ sudo shutdown -k now
```

「shutdown -k now」で、シャットダウンのテストが可能
※実際にシャットダウンはされない

We trust you have received the usual lecture from the local System Administrator. It usually boils down to these three things:

:

```
[sudo] password for testuser:
```

実行ユーザーのパスワードを入力
※rootユーザーのパスワードではない
※一度入力すると一定時間は再入力不要

```
Broadcast message from root@centos6  
(/dev/pts/0) at 0:40 ...
```

```
The system is going down for maintenance NOW!
```



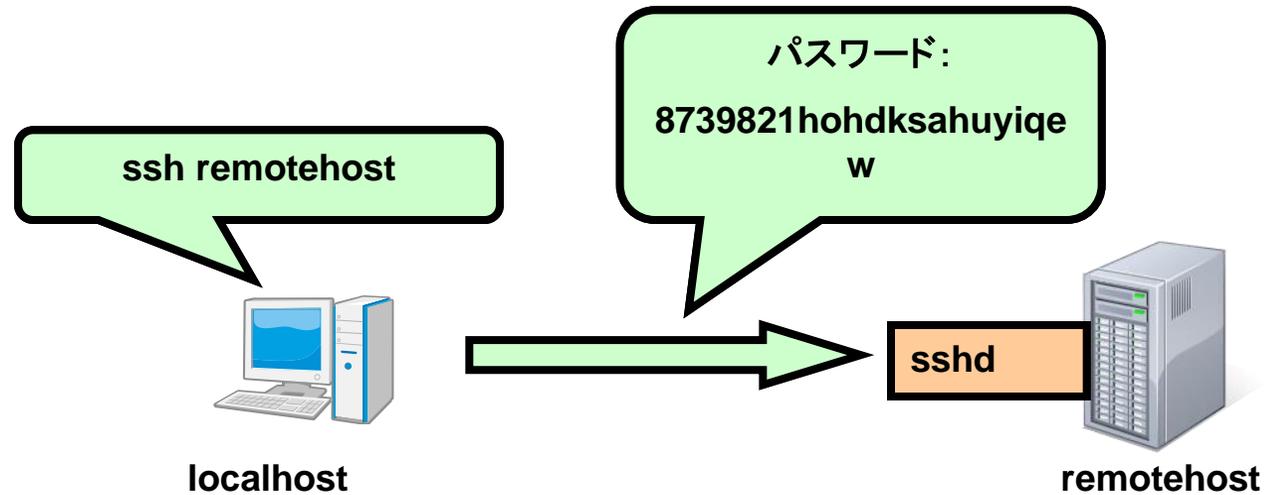
これ以降はお時間があれば・ ・



暗号化によるデータの保護（主題110）

■ ssh (Secure Shell)

リモートホストに接続するためのプログラム。ネットワーク上を流れるデータは暗号化される。





■SSHでは既定で、2種類の認証をサポート

- ホスト認証：接続先のホストのなりすましを防ぐ
- クライアント認証：信頼できる接続元であるかを確認
 - 公開鍵認証
 - パスワード認証

■暗号化方式

■共通鍵方式

暗号化も復号化も同じ鍵（共通鍵）

■公開鍵方式

暗号化と復号化を異なる鍵（公開鍵と秘密鍵）



暗号化によるデータの保護 (主題110)

■ユーザー認証における公開鍵認証の有効化

```
[root@localhost ~]# ssh-keygen
```

ユーザー認証用の鍵を生成: ~/.sshに生成

```
[root@localhost ~]# scp .ssh/id_dsa.pub remotehost:/root
```

scpコマンドで接続先に公開鍵を転送

```
[root@remotehost ~]# cat id_dsa.pub >> .ssh/authorized_keys
```

```
[root@localhost ~]# ssh remotehost
```

所定のファイルに格納

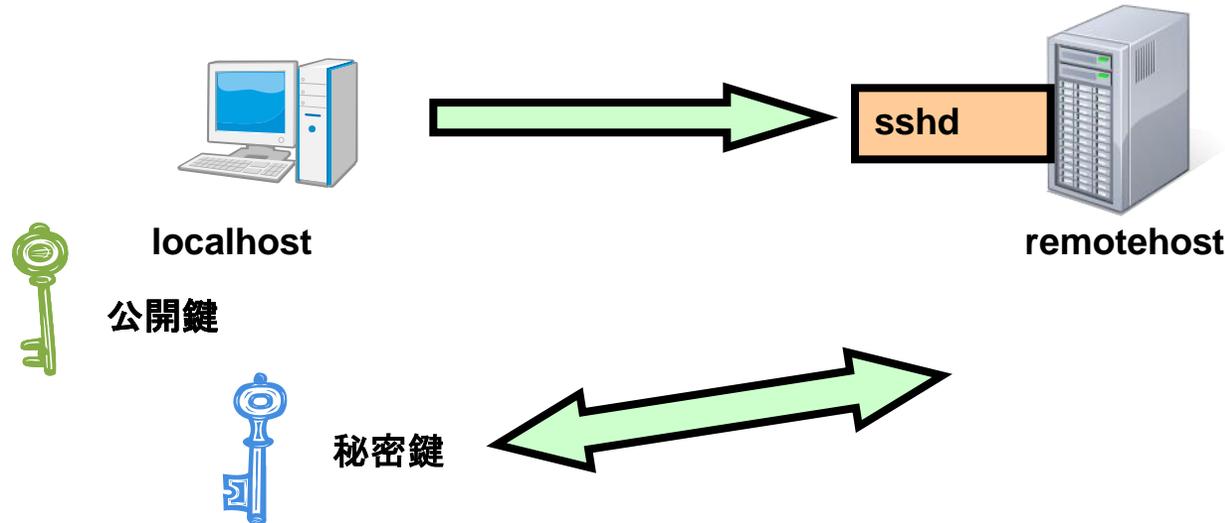
```
[root@remotehost ~]#
```

```
[root@remotehost ~]# vi /etc/ssh/sshd_config
```

```
PasswordAuthentication no
```

```
[root@remotehost ~]# service sshd restart
```

パスワード認証を無効化し、事前に公開鍵を登録したユーザーのみ接続できるようにする





ご静聴ありがとうございました！