



株式会社 ケイ・シー・シー



Linux Professional Institute Japan

**LPI-JAPAN**

# LPICレベル2 技術解説無料セミナー

株式会社ケイ・シー・シー

西日本センターユニット ITラーニングセンター

福田 浩之



## ■ 会社概要

株式会社ケイ・シー・シー

<http://www.kcc.co.jp/>

## ■ 講師紹介

株式会社ケイ・シー・シー

西日本センターユニット ITラーニングセンター

福田 浩之

Linuxをはじめ、ネットワーク・セキュリティ・VoIPなど様々な分野の研修を担当し、幅広い知識をバックボーンとした説得力ある講習会を実施。

注目度の高いHTML5、Android、iOSなど最新技術の研修も担当し、最新情報を取り入れた講習会を心掛けています。

双方向になるようコミュニケーションを重視した研修は、高い評価を得ている。



## 1. LPICレベル2 試験概要

- LPIC試験概要
- Linux学習環境の構築
- 学習方法

## 2. 技術解説項目

### 201試験範囲より

- 主題202 システムの起動
- 主題203 ファイルシステムとデバイス

### 202試験範囲より

- 主題212 システムのセキュリティ
  - セキュアシェル(SSH)



株式会社 ケイシーシー

# LPICレベル2 試験概要



# 201試験の出題範囲



株式会社ケイシーシー

- 主題200: キャパシティプランニング
- 主題201: Linuxカーネル
- 主題202: システムの起動
- 主題203: ファイルシステムとデバイス
- 主題204: 高度なストレージ管理
- 主題205: ネットワーク構成
- 主題206: システムの保守



# 202試験の出題範囲



- 主題207: ドメインネームサーバ
- 主題208: HTTPサービス
- 主題209: ファイル共有
- 主題210: ネットワーククライアントの管理
- 主題211: 電子メールサービス
- 主題212: システムのセキュリティ



## ■ インターネットをフルに活用

- 関連キーワードで分からないものほとにかく調べる
- 信頼できる「お気に入りサイト」を見つけておく
  - JM Project, Linux JF Project, @ITなど

## ■ 実機を使った学習

- コマンドは実機で実行してみる
- manを活用する

## ■ 学習環境の構築

- 無償ディストリビューション(CentOS, Fedora, Ubuntu等)を利用
- Linux専用マシンがあればベスト
- VM環境の構築を検討
  - VMWare / Virtual Boxなど無償仮想化ツールの導入



## ■ 幅広い出題範囲

- 出題範囲詳細をもとにして、すべて網羅する
- 得意分野をつくる

## ■ 実務に則した問題

- 参考書だけの勉強ではなく、実機で確認する
- コマンドの出力結果やエラーメッセージをしっかり見ておく
- 重要な設定ファイルは主な設定項目(パラメータ)も覚える



## ■ CBT (Computer Based Testing) 試験

- コンピュータを操作して問題に解答
- 試験中、問題は何度も繰り返し参照可能
- 試験終了と同時に結果が判明

## ■ 試験時間の有効活用

- 90分で約60問の問題
- 四者択一または五者択一、複数選択、記入式の3パターン
  - 問題はしっかり読む
  - あやふやな問題はチェックをつけて、後から解答する
  - 全体的に見直す時間を確保する



# 技術解説

## ■ 主題202 システムの起動

- 202.1 システムの起動をカスタマイズする
- 202.2 システムのリカバリ



## ■Linuxの起動処理

1. 電源ONによりBIOSが起動する
2. BIOSからブートローダーが呼び出される
3. ブートローダーがカーネルを起動する
4. カーネルがinitプロセスを起動する

initがLinuxの各種デーモンやアプリケーションサービスの起動をつかさどるが、古くから利用されている「Sysvinit」、Sysvinitの代替えとして「Upstart」、そして最新の「systemd」がある。





## ■ SysVinitによる起動までの流れ

①電源ON

②BIOSプログラムの起動

③ブートローダの起動

④カーネルの起動

⑤initプロセスの起動

⑥rcスクリプトの起動

⑦ランレベルごとのサービス起動

⑧ログイン画面の表示

•ハードウェアのチェック・初期化  
•起動デバイスの選択

•起動するカーネルを選択

•メモリ・ハードウェアの初期化

•最初に起動するプロセス

•システムの初期化

•ランレベルに応じたサービス起動



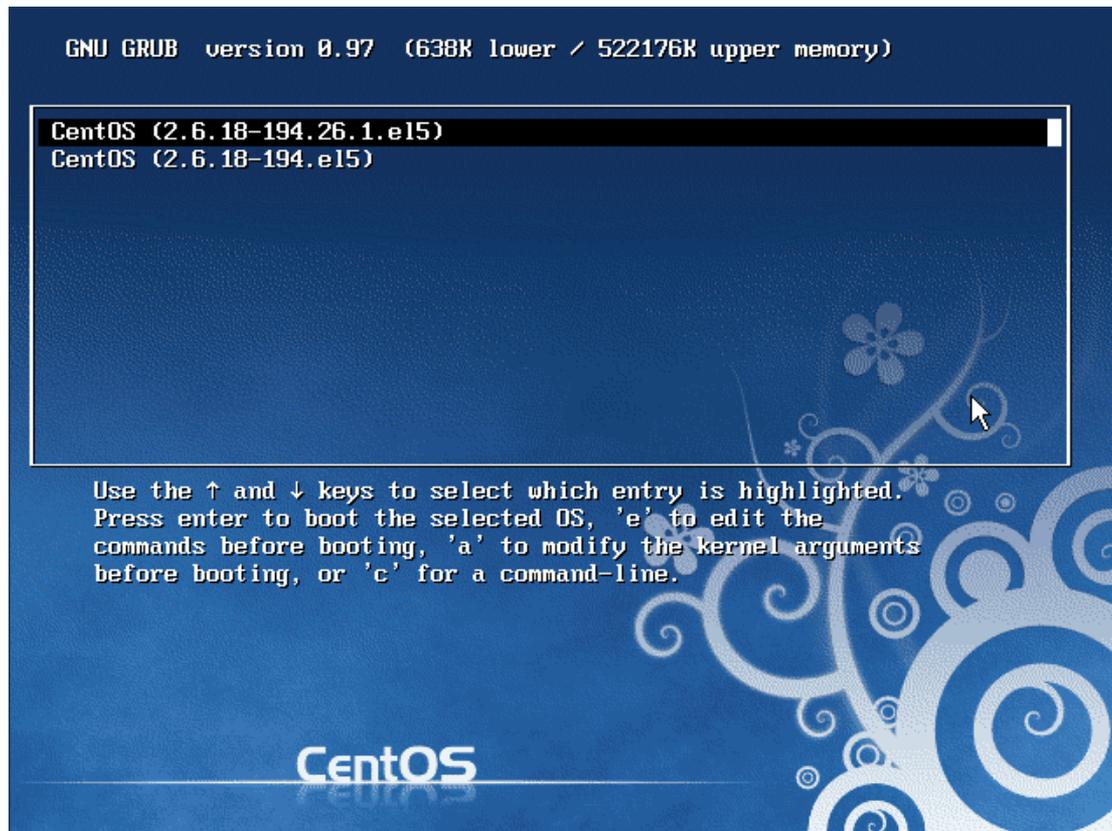
## ■ BIOSプログラムとは

- 各種ハードウェアの調整と初期化
- 起動デバイスのチェックを行い、システムのブートデバイスを探す
  - 一般的にはFDD→CD-ROM→HDDの順番でデバイスを検索
  - 指定されたブートデバイスが存在しない場合は次のデバイスを検索
  - デバイスの先頭領域に格納されたブートローダを実行



## ■ブートローダとは

- OSをディスクから読み出して起動するプログラム
- 複数のブートローダが存在する場合は、起動するカーネルを選択





## ■ GRUB

- 現在主流で使用されているブートローダ

## ■ GRUB設定ファイル/boot/grub/grub.conf

```
# cat /boot/grub/grub.conf
default=0
timeout=5
splashimage=(hd0,0)/grub/splash.xpm.gz
hiddenmenu
title CentOS (2.6.18-194.26.1.el5)
    root (hd0,0)
    kernel /vmlinuz-2.6.18-194.26.1.el5 ro root=LABEL=/ rhgb quiet
    initrd /initrd-2.6.18-194.26.1.el5.img
```

### ○主な設定パラメータ

default	デフォルトで起動するカーネル番号
timeout	入力待ちタイムアウト時間(秒単位)
splashimage	メニュー表示時の背景画面
hiddenmenu	カーネル選択メニューを非表示

title	メニューに表示するカーネルエントリ名
root	OSが格納されているパーティションの指定
kernel	カーネルイメージとカーネルに渡す引数
initrd	イニシャルRAMディスクの指定



## ■ GRUB2

- GRUB(ver1)を改良したブートローダ
- 設定ファイルは、`/boot/grub/grub.cfg`
  - ※但し、GRUB Legacyとは異なり、エディタを使って直接編集しない

### <設定ファイルの更新手順>

- ① `/etc/default/grub` ファイルを編集する
- ② `update-grub` (または、`update-grub2`) コマンドを実行する
- ③ `/boot/grub/grub.cfg` が生成される



## ■カーネルの主な役割

- OS上で周辺機器・CPU・メモリなどを制御
- アプリケーションの実行環境を整備

## ■カーネルの起動

- 設定ファイルの記述に従ってOSに必要なプロセスが呼び出される
  - ハードウェア・メモリのチェック
  - ルートファイルシステムをマウント
  - initプログラムを実行



## ■initプロセスの起動

- カーネルによって一番最初に起動されるプロセス
- /etc/inittabの設定に従い、OSに必要なプロセスを起動する

## ■ランレベル

- システムの状態を表す値

0	システムの停止
1, s, S	シングルユーザーモード
2	NFSファイル共有のないマルチユーザーモード
3	完全マルチユーザーモード(テキストベース)
4	未使用
5	完全マルチユーザーモード(X Window System)
6	システムの再起動



## ■/etc/inittab

### <書式> ID:ランレベル:アクション指示子:実行される動作

```
id:5:initdefault:

# System initialization.
si::sysinit:/etc/rc.d/rc.sysinit

I0:0:wait:/etc/rc.d/rc 0
I1:1:wait:/etc/rc.d/rc 1
I2:2:wait:/etc/rc.d/rc 2
I3:3:wait:/etc/rc.d/rc 3
I4:4:wait:/etc/rc.d/rc 4
I5:5:wait:/etc/rc.d/rc 5
I6:6:wait:/etc/rc.d/rc 6

...

1:12345:respawn:/sbin/mingetty tty1
2:2345:respawn:/sbin/mingetty tty2
3:2345:respawn:/sbin/mingetty tty3
4:2345:respawn:/sbin/mingetty tty4
5:2345:respawn:/sbin/mingetty tty5
6:2345:respawn:/sbin/mingetty tty6

x:5:respawn:/etc/X11/prefdm -nodaemon
```

#### ○主な設定項目

id:5:initdefault:

- デフォルトランレベルを5に設定

I5:5:wait:/etc/rc.d/rc 5

- ランレベルを引数にしてrcスクリプトを実行

1:12345:respawn:/sbin/mingetty tty1

- 仮想端末の立ち上げ

x:5:respawn:/etc/X11/prefdm -nodaemon

- X Windows Systemの起動



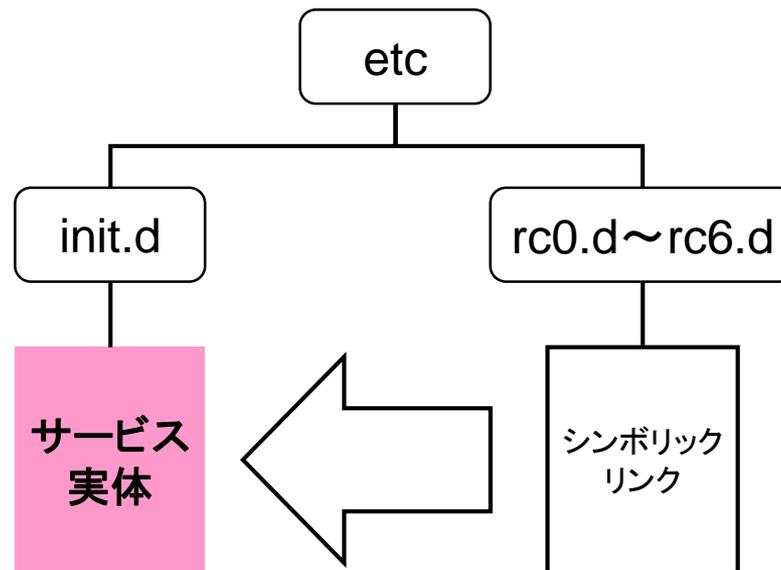
## ■システムの初期化

- カーネルパラメータ
- システムクロック
- キーボード配置
- コンソールフォント
- ホスト名
- USBコントローラ
- ルートファイルシステムのチェックと読み書き可能での再マウント
- カーネルモジュール
- ルートファイルシステム以外のファイルシステムのチェックとマウント
- スワップ領域



## ■ /etc/rc

- 起動/停止するサービスは/etc/rcX.dディレクトリ内のファイルで設定  
(RedHat系のディストリビューションは/etc/rc.d/rcX.dディレクトリ)
- サービス実体は/etc/init.dディレクトリに存在



起動/停止する  
サービスの一覧  
(ランレベルごとに  
分かれています)



## ■ /etc/rcX.dディレクトリ(X=0~6)

〈書式〉 /etc/init.d/スクリプトファイル名 アクション

- Kが停止(Kill)、 Sが開始(Start)を表す
- 最初にサービス停止を実行、次にサービス起動を実行

```
# ls /etc/rc5.d
```

```
K01dnsmasq           K35vncserver         K89rdisc             S18rpcidmapd         S90crond
K02NetworkManager   K35winbind           K91capi              S19rpcgssd           S90xfstools
K02avahi-dnssconfd   K36postgresql        S00microcode_ctl    S22messagebus        S95anacron
K02oddjobd           K50ibmasm            S02lvm2-monitor     S23setroubleshoot   S95atd
K05conman            K50netconsole        S04readahead_early  S25bluetooth         S97yum-updatesd
K05innd              K50tux               S05kudzu            S25netfs             S98avahi-daemon
K05saslauthd         K50vsftpd            S08ip6tables        S25pcscd             S99firstboot
K05wdaemon           K69rpcsvcgssd        S08iptables         S26acpid             S99local
K10dc_server         K73ypbind            S08mcstrans         S26apmd              S99smartd
K10psacct            K74nscd              S09isdnsd           S26haldaemon
K10tcsd              K74ntpd              S10network          S26hidd
K12dc_client         K80kdump              S11auditd           S28autofs
K15httpd             K85mdmpd             S12restorecond      S50hpplip
K20nfs               K87multipathd        S12syslog           S55sshd
K20rwhod             K87named             S13cpuspeed         S56cups
K24irda              K88wpa_supPLICANT    S13irqbalance       S56rawdevices
K25squid             K89dum                S13portmap          S56xinetd
K30spamassassin     K89netplugd          S14nfslock          S80sendmail
K35smb               K89pand              S15mdmonitor        S85gpm
```



## ■ サービスの自動起動、停止制御

- RedHat系                   chkconfig コマンド
- Debian系                   update-rc.dコマンド
- openSUSE                   insservコマンド



## ■ サービスの手動起動、停止制御

〈書式〉 /etc/init.d/スクリプトファイル名 アクション

○vsftpdサービスの状態を確認する

```
# /etc/init.d/vsftpd status  
vsftpd は停止しています
```

○vsftpdサービスを起動する

```
# /etc/init.d/vsftpd start  
vsftpd を起動中: [ OK ]
```

○vsftpdサービスを停止する

```
# /etc/init.d/vsftpd stop  
vsftpd を停止中: [ OK ]
```

〈主なアクション〉

start	サービス起動
stop	サービス停止
restart	サービス再起動
status	サービス状態確認



- 初期ランレベルによりログイン画面の提供プログラムが異なる
  - ランレベル3 (CUIログイン)
    - loginプログラム
  - ランレベル5 (GUIログイン)
    - ディスプレイマネージャ (gdm, kdmなど)



## ■ init コマンド

〈書式〉 `init ランレベル`

- 指定したランレベルに変更する

## ■ telinit コマンド

- /sbin/init へのシンボリックリンク (実行時に init コマンドを参照)



## ■systemd とは

- Sysvinit や Upstart にかわる新しい仕組み
- システム状態を表すランレベルは使用しない

## ■systemd の仕組み

- 各種サービスは「**ユニット**」という単位で管理する
- ユニットは役割によって「**タイプ**」に分かれている

タイプ	役割
service	各種デーモンやサービスの起動
target	起動プロセスやサービスなどの複数のユニットをグループにしてまとめたもの
mount	ファイルシステムのマウントポイント制御
device	ディスクデバイス



## ■ systemd の仕組み(つづき)

- 起動プロセスやサービスなどの複数のユニットをグループとしてまとめたものを「**ターゲット**」という

例) ネットワークに関する処理のユニットをグループ化したもの ⇒ network.target

システムの状態	systemd target	ランレベル
停止	poweroff.target	0
シングルユーザーモード	rescue.target	1
完全マルチユーザーモード(テキストベース)	multi-user.target	2、3、4
完全マルチユーザーモード(X Window System)	graphical.target	5
システムの再起動	reboot.target	6
緊急モード	emergency.target	-



## ■ systemctl とは

ユニットを制御するためのコマンド

<書式> **systemctl** [オプション] コマンド

[動作モード関連のコマンド]

コマンド	内容
get-default	デフォルトのモードを確認する
set-default ターゲット	デフォルトのモードを変更する
isolate ターゲット	現在のモードを変更する
default	デフォルトのモードにする
rescue	レスキューモードに入る
emergency	緊急モードに入る
halt	シャットダウンして終了する
reboot	シャットダウンして再起動する



## ■ デフォルトターゲットの確認／設定する

システムを起動すると default.target (設定ファイル) に指定されたターゲットが起動される

○ default.target の設定を確認する

```
# systemctl get-default  
graphical.target
```

○ システム起動時のターゲットを multi-user.target に設定する

```
# systemctl set-default multi-user.target  
rm '/etc/systemd/system/default.target'  
ln -s '/usr/lib/systemd/system/multi-user.target'  
'/etc/systemd/system/default.target'
```

## ■ 現在の動作モードを変更する

○ 完全マルチユーザーモード(テキストベース)に変更する

```
# systemctl isolate multi-user.target
```



## ■ systemctl コマンドでレスキューモードや緊急モードに変更する

### ○ レスキューモードに変更する

```
# systemctl rescue
```

または

```
# systemctl isolate rescue.target
```

### ○ 緊急モードに変更する

```
# systemctl emergency
```

または

```
# systemctl isolate emergency.target
```

※レスキューモード、緊急モードの詳細は 35頁参照



### ■ ユニットの状態を確認する

○ vsftpd サービスの状態を確認する

```
# systemctl status vsftpd.service
```

### ■ ユニットを起動する

○ vsftpd サービスを起動する

```
# systemctl start vsftpd.service
```

### ■ ユニットを停止する

○ vsftpd サービスを停止する

```
# systemctl stop vsftpd.service
```

[ユニット制御関連のコマンド]

status	サービス状態確認
start	サービス起動
stop	サービス停止
restart	サービス再起動

※ ユニット名の「.service」拡張子は省略できる



## ■各ユニットの設定ファイル格納場所

<code>/usr/lib/systemd/system/</code>	パッケージが提供するシステム標準(デフォルト)の設定ファイルを格納
<code>/etc/systemd/system/</code>	システム管理者が設定変更したファイルを格納 デフォルトの設定を変更する場合は、当ディレクトリにユニットファイルをコピーして編集する

両方のディレクトリに同名の設定ファイルがある場合は、`/etc/systemd/system/`のファイルが優先



# 技術解説

## ■ 主題202 システムの起動

- 202.1 システムの起動をカスタマイズする
- 202.2 システムのリカバリ



## ■レスキューモード

- リカバリCR-ROMやインストールメディアを利用してシステムを起動する

## ■シングルユーザーモード

- デフォルトの起動モードの代わりに、rootユーザーのシェルが起動し作業が行える

## ■緊急モード ※このモードはsystemdのみ

- 従来のレスキューモードよりも、さらに深刻な場合に利用する起動モード。ルート(/)ファイルシステムすらマウントできない場合などに利用する



## ■ レスキューモード

- ハードディスクから起動できなくなった場合、代わりにDVD-ROMから起動するモード

## ■ レスキューモードの実行手順

### ① DVD-ROMから起動

※事前にBIOSでデバイス起動順序の設定が必要

### ② ブートプロンプトに「linux rescue」と入力

```
boot: linux rescue
```

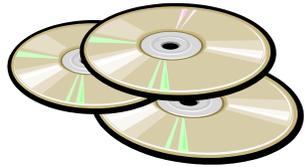
### ③ 言語選択とキーボード選択を行う

※日本語は表示できないため注意

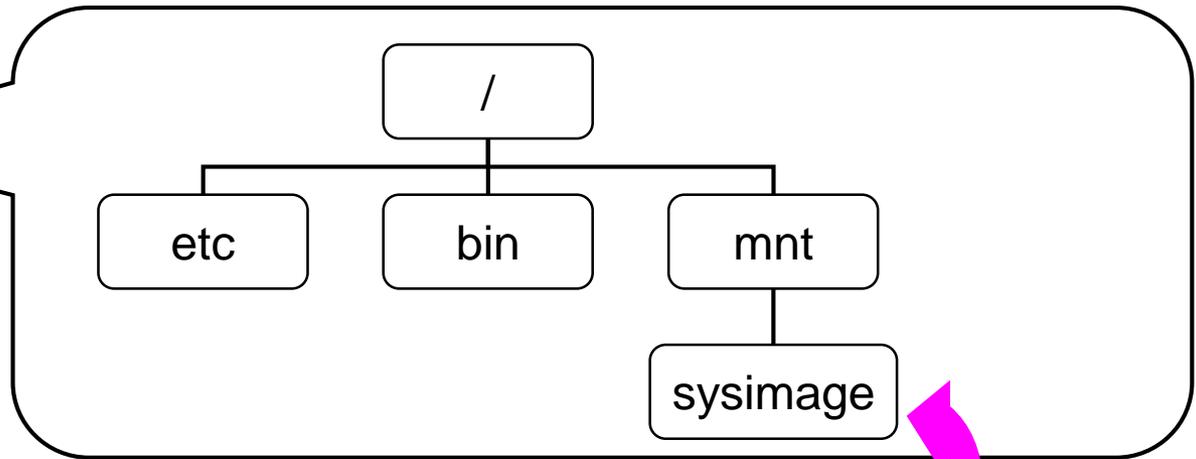


## ■レスキューモードの環境

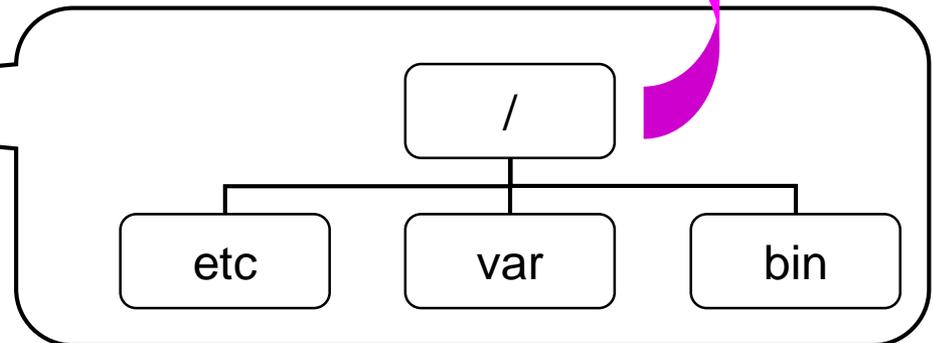
- メモリ上にDVD-ROMのイメージが展開される
- 全てのコマンドが使用できるわけではない  
(RAMディスクのサイズ制限のため)



メモリ上に展開されるイメージ



修復対象ディスクの内容



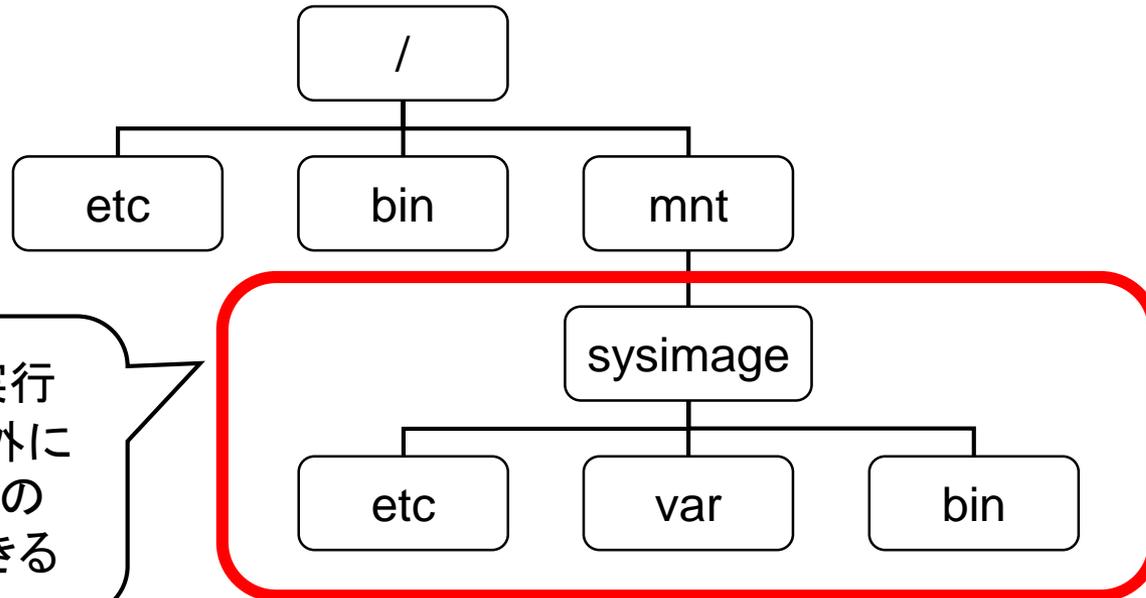


## ■ chroot コマンド

- ルートディレクトリを変更する

```
# chroot /mnt/sysimage
```

DVD-ROMブートされたシステムの  
ルートディレクトリ



chrootコマンドを実行することにより、枠外にあるディレクトリへのアクセスを制限できる



## ■ シングルユーザーモード

- ① GRUB起動メニューが表示されたら何かキーを押す
- ② 「e」を押す
- ③ <sysVinit/Upstartの場合>
  - a. カーネルの最後に「**△S**」を追加(△は半角空白)
  - b. 「Enter」でメニュー画面に戻り、「b」を押す<systemdの場合>
  - a. **systemd.unit=rescue.target** を追加
  - b. 「Enter」でメニュー画面に戻り、「Ctrl」+「x」を押す
- ④ シングルユーザーモードで起動する



## ■ 緊急モード

※このモードはsystemdのみ

- ① GRUB起動メニューが表示されたら何かキーを押す
- ② 「e」を押す
- ③ `systemd.unit=emergency.target` を追加
- ④ 「Enter」でメニュー画面に戻り、「Ctrl」+「x」を押す



## ■ブートローダのインストール

- デュアルブート環境の作成やブートローダ破損時に実行
- grub-installコマンド
  - GRUBを強制的にインストール
  - 次回起動時に設定が反映される



# 技術解説

## ■ 主題203 ファイルシステムとデバイス

- 203.1 Linuxファイルシステムを操作する
- 203.2 Linuxファイルシステムの保守
- 203.3 ファイルシステムを作成してオプションを構成する



# ファイルシステムの操作

①パーティションの作成

fdiskコマンド、partedコマンド

②ファイルシステムの作成

mke2fsコマンド、mkfsコマンド

③ファイルシステムチェック

e2fsckコマンド、fsckコマンド

④マウント

mountコマンド、umountコマンド

⑤自動マウントの設定

/etc/fstabファイル



## ②ファイルシステムの作成

### ■ mke2fs コマンド

〈書式〉 mke2fs オプション デバイスファイル名

○ /dev/sda6 上に ext3 ファイルシステムを作成する

```
# mke2fs -j /dev/sda6
```

○ 主なオプション

-j	ext3ファイルシステムを作成する
----	-------------------



## ②ファイルシステムの作成



### ■mkisofsコマンド

<書式> mkisofs [オプション] ディレクトリ名

- CD-ROMなどに用いられるISO9660ファイルシステムを作成

○/etcのISO9660イメージを/tmp/etc.isoとして作成する

```
# mkisofs -o /tmp/etc.iso /etc
```



## ③ファイルシステムのチェック

### ■ fsck コマンド

〈書式〉 `fsck` [オプション] [デバイス名]

- ファイルシステムの整合性をチェックする

### ○ 主なオプション

-r	対話的に修復を実行する
-t タイプ	ファイルシステムの種類を指定する
-A	/etc/fstabに記述されている全ファイルシステムに対して実行する
-N	実行せず、実行するとどうなるかのみ表示する



# ③ファイルシステムのチェック

## ■ e2fsck コマンド

〈書式〉 **e2fsck** [オプション] [デバイス名]

- ext2, ext3ファイルシステムの整合性をチェックする。

## ○ 主なオプション

-b ブロック	指定したスーパーブロックのバックアップを使って修復する
-c	不良ブロックをチェックする
-f	ファイルシステムの状態がcleanでもチェックする
-p	全ての不良ブロックを自動的に修復する

## コマンド実行時の注意点

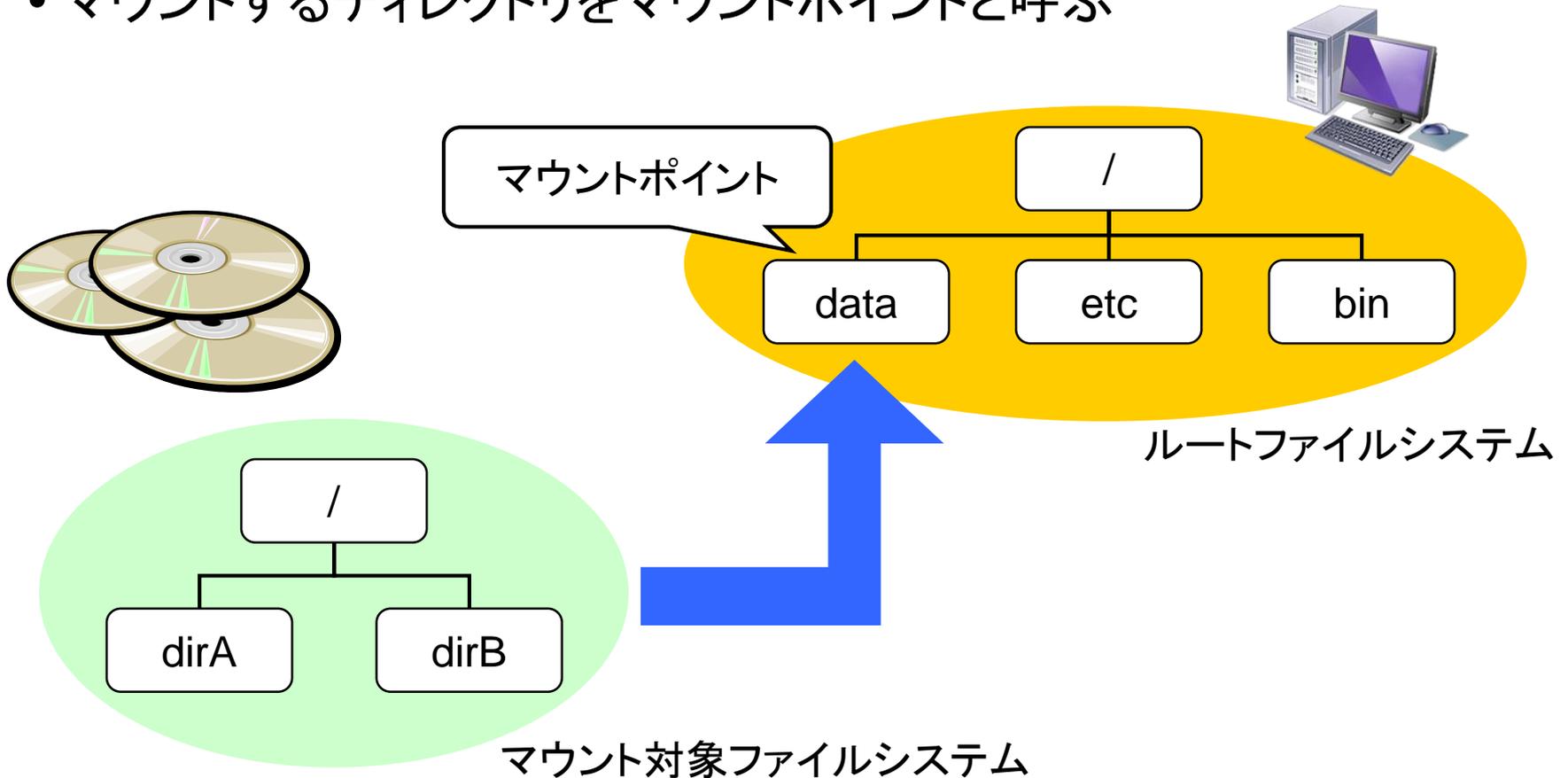
- 対象となるファイルシステムをアンマウントする
- ルートファイルシステムをチェックする場合は以下のいずれかの方法をとる
  - DVD-ROMからブートしてチェック
  - shutdownコマンドに「-F」オプションをつけて実行



# ④ファイルシステムのマウント

## ■マウント

- ファイルシステムをOSに認識させ、使用可能にすること
- マウントするディレクトリをマウントポイントと呼ぶ





# ④ファイルシステムのマウント



## ■mountコマンド

<書式> mount [オプション] デバイスファイル名 マウントポイント

※/etc/fstabに記述がある場合はマウントポイントのみでも可

○/dev/sda2上にあるext3ファイルシステムを/homeにマウントする

```
# mount -t ext3 /dev/sda2 /home
```

## ○主なオプション

-a	/etc/fstabで指定されているファイルシステムを全てマウントする	
-o オプション	-o remount	再マウント
	-o noexec	バイナリの実行を許可しない
-t タイプ	ファイルシステムの種類を指定する	

## ■umountコマンド

<書式> umount [オプション] デバイスファイル名 or マウントポイント



## ■ /etc/fstab

マウントするファイルシステムに関する情報を格納する設定ファイル

### <書式>

デバイスファイル名 マウントポイント ファイルシステム種類 マウントオプション ダンプ fsck順序

```
# cat /etc/fstab
/dev/sda1      /boot        ext3          defaults      1            2
LABEL=/       /            ext3          defaults      1            1
tmpfs         /dev/pts    tmpfs         defaults      0            0
devpts       /dev/pts    devpts        gid=5,mode=620 0            0
sysfs        /sys        sysfs         defaults      0            0
proc         /proc       proc          defaults      0            0
/dev/sda3    swap        swap          defaults      0            0
```



## ■ 主なマウントオプション

defaults	デフォルトオプション (async, auto, dev, exec, nouser, rw, suid)
async	ファイルシステムに対する全ての入出力を非同期で行う
sync	ファイルシステムに対する全ての入出力を同期で行う
auto	mount -aを実行したときにマウントする
noauto	mount -aを実行したときにマウントしない
dev	ファイルシステム上のデバイスファイルを使用できる
exec	バイナリの実行を許可する
noexec	バイナリの実行を禁止する
user	一般ユーザのマウントを許可し、マウントしたユーザのみアンマウントできる
users	一般ユーザのマウントを許可し、マウントしたユーザ以外でもアンマウントできる
nouser	一般ユーザのマウントを禁止する
ro	読み出し専用でマウントする
rw	読み書きを許可してマウントする
suid	SUID,SGIDビットを有効にする
nosuid	SUID,SGIDビットを無効にする



## ■ /etc/mtab

- 現在マウントされているファイルシステムを表示

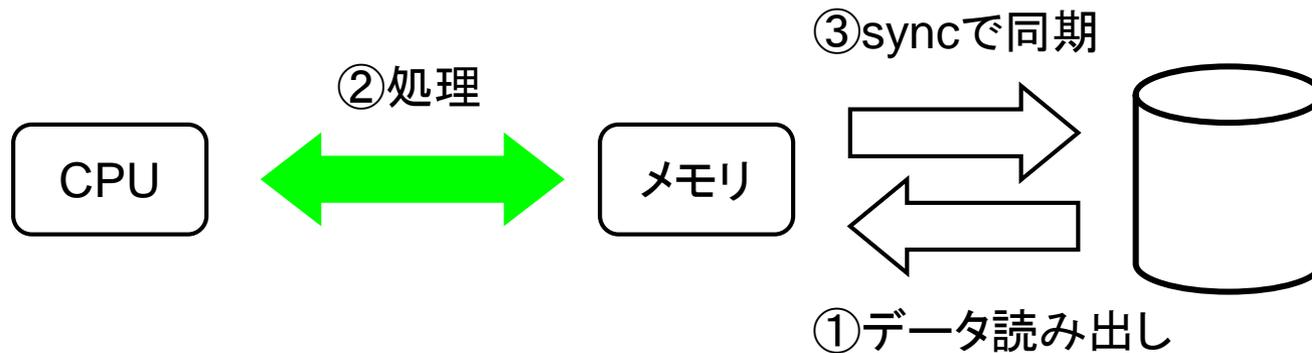
## ■ /proc/mounts

- /etc/mtabとほぼ同じ内容



## ■ sync コマンド

- ディスクバッファ領域にあるデータをディスクに書き込む





- スワップとは
  - ブロックデバイス上の仮想的なメモリ領域
- スワップ領域の作成
  - mkswapコマンド
- スワップ領域の有効化／無効化
  - swaponコマンド／swapoffコマンド
- アクティブなスワップ領域を表示
  - swapon -s  
(内容は/proc/swapsと同じ)



## ■ dumpe2fs コマンド

- スーパーブロック情報を確認する

```
# dumpe2fs /dev/sda1
dumpe2fs 1.39 (29-May-2006)
Filesystem volume name:   /boot
Last mounted on:         <not available>
Filesystem UUID:         73359549-22fb-4320-a0a9-08383e8285c0
Filesystem magic number: 0xEF53
Filesystem revision #:   1 (dynamic)
Filesystem features:     has_journal ext_attr resize_inode dir_index ...
...

Group 0: (Blocks 1-8192)
  Primary superblock at 1, Group descriptors at 2-2
  Reserved GDT blocks at 3-258
  Block bitmap at 259 (+258), Inode bitmap at 260 (+259)
  Inode table at 261-511 (+260)
  0 free blocks, 1984 free inodes, 2 directories
  Free blocks:
  Free inodes: 25-2008
Group 1: (Blocks 8193-16384)
  Backup superblock at 8193, Group descriptors at 8194-8194
...
```



## ■ tune2fs コマンド

〈書式〉 tune2fs [オプション] デバイスファイル名

- ファイルシステムのチューニングを行う

○ ext2 ファイルシステムを ext3 ファイルシステムに変更

```
# tune2fs -j /dev/sda6
```

○ 主なオプション

-j	ジャーナル機能をファイルシステムに追加する
----	-----------------------



## ■ Btrfsとは

Linux向けの新しいファイルシステム。ZFS\*<sup>1</sup>ファイルシステムを元に開発された。Btrfsは「スナップショット」の機能を持ち、耐障害性に優れている。

スナップショットはサブボリューム\*<sup>2</sup>単位で取得できる。

※1 ZFS:オラクルの solalis 上で使用されているファイルシステム

※2 サブボリューム:ファイルシステムをさらに細かく分割する単位

### ○サブボリュームを作成する

```
# btrfs subvolume create /mnt/sv1  
                        【サブボリューム名】
```

### ○ext2、ext3またはext4ファイル・システムをbtrfsに変換する

```
# btrfs-convert /dev/sdb  
                        【デバイス名】
```



## ■systemctl コマンド

- ユニットの状態を確認する

〈書式〉 systemctl アクション ユニット名

※ユニット名の「.service」拡張子は省略できる

### ○vsftpdサービスの状態を確認する

```
# systemctl status vsftpd
```

### ○vsftpdサービスを起動する

```
# systemctl start vsftpd
```

### ○vsftpdサービスを停止する

```
# systemctl stop vsftpd
```

### 〈主なアクション〉

start	サービス起動
stop	サービス停止
restart	サービス再起動
status	サービス状態確認



# 技術解説

## ■ 主題212 システムのセキュリティ

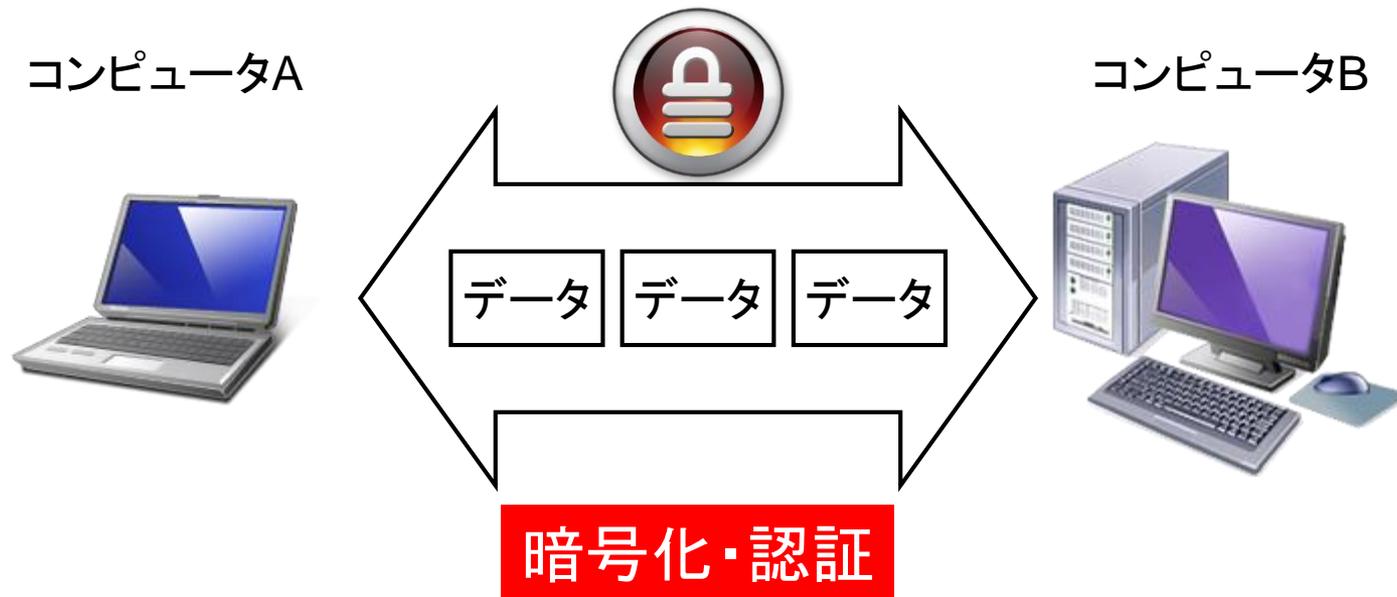
- 212.3 セキュアシェル(SSH)



# セキュアシェル (SSH)

## ■ ssh

- 認証と暗号化によりリモート操作を安全に行うシェル





## ■ SSH (Secure SHell)

- 2種類のバージョン (SSHv1, SSHv2)
- 2種類の暗号アルゴリズムをサポート
  - RSA (SSHv1, SSHv2)
  - DSA (SSHv2)

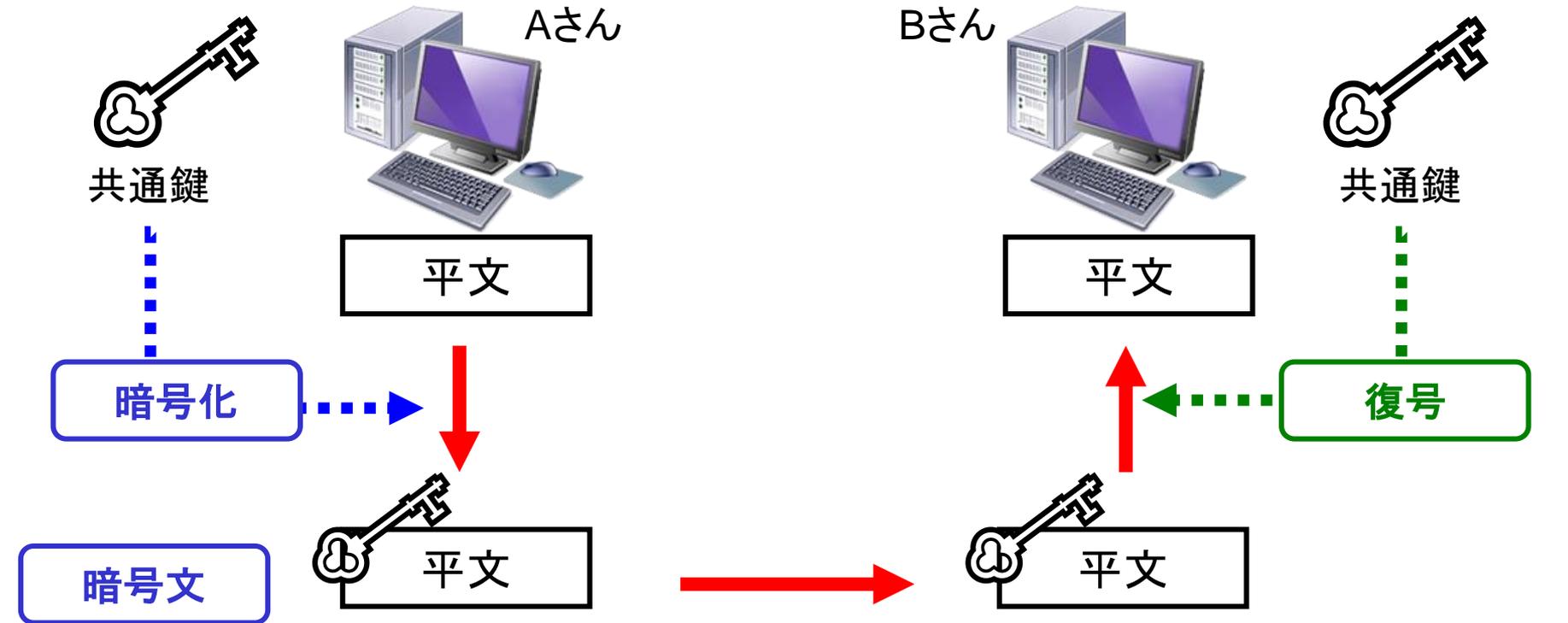
○ 公開鍵ファイル名 (サーバ側でsshd起動時、/etc/ssh直下に自動生成)

	秘密鍵	公開鍵
SSHv1 RSA	ssh_host_key	ssh_host_key.pub
SSHv2 DSA	ssh_host_dsa_key	ssh_host_dsa_key.pub
SSHv2 RSA	ssh_host_rsa_key	ssh_host_rsa_key.pub

- 複数の認証方式をサポート
  - ホスト認証
  - ユーザ認証 (パスワード認証、ホストベース認証、公開鍵認証)
- ポートフォワーディング機能



# 暗号方式(共通鍵暗号方式)



**メリット**

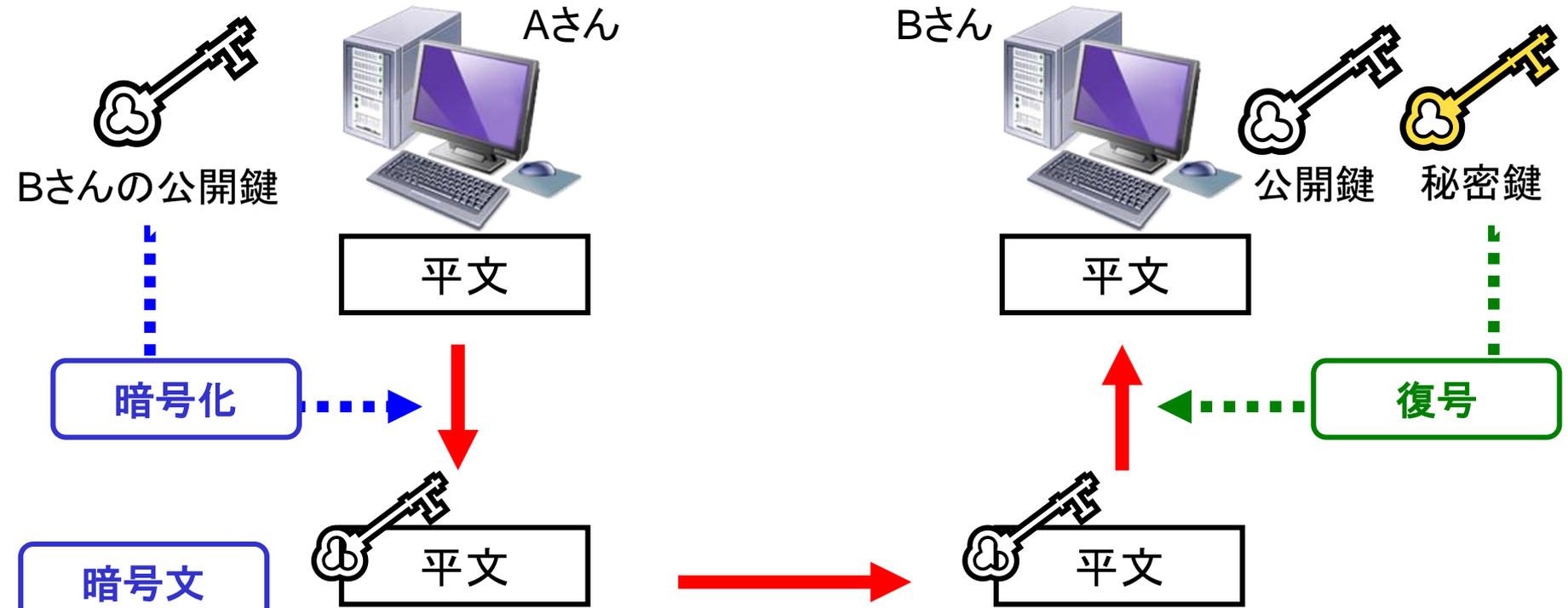
- 暗号化・復号処理が高速

**デメリット**

- 鍵の秘密配送が必要
- 通信相手ごとに共通鍵が必要



# 暗号方式(公開鍵暗号方式)



**メリット**

- 鍵の秘密配送が不要
- 鍵管理が容易に

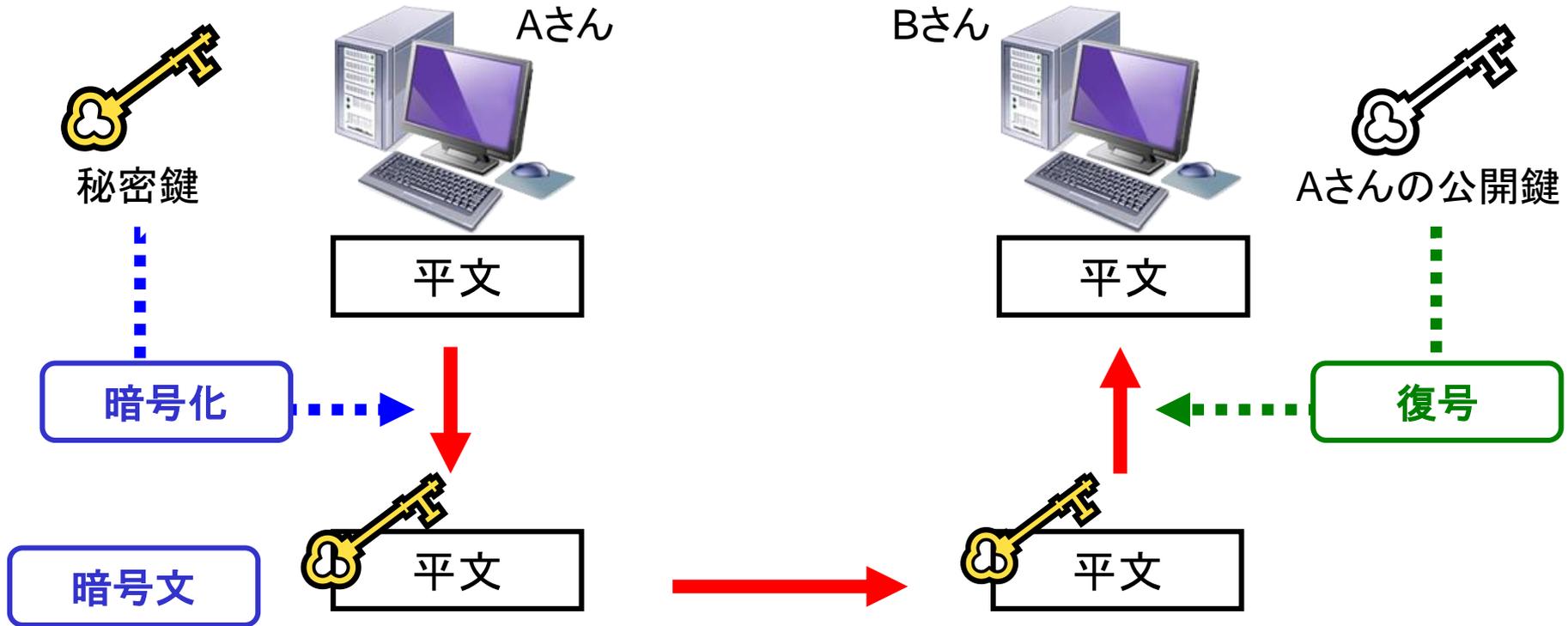
**デメリット**

- 暗号化・復号処理が低速



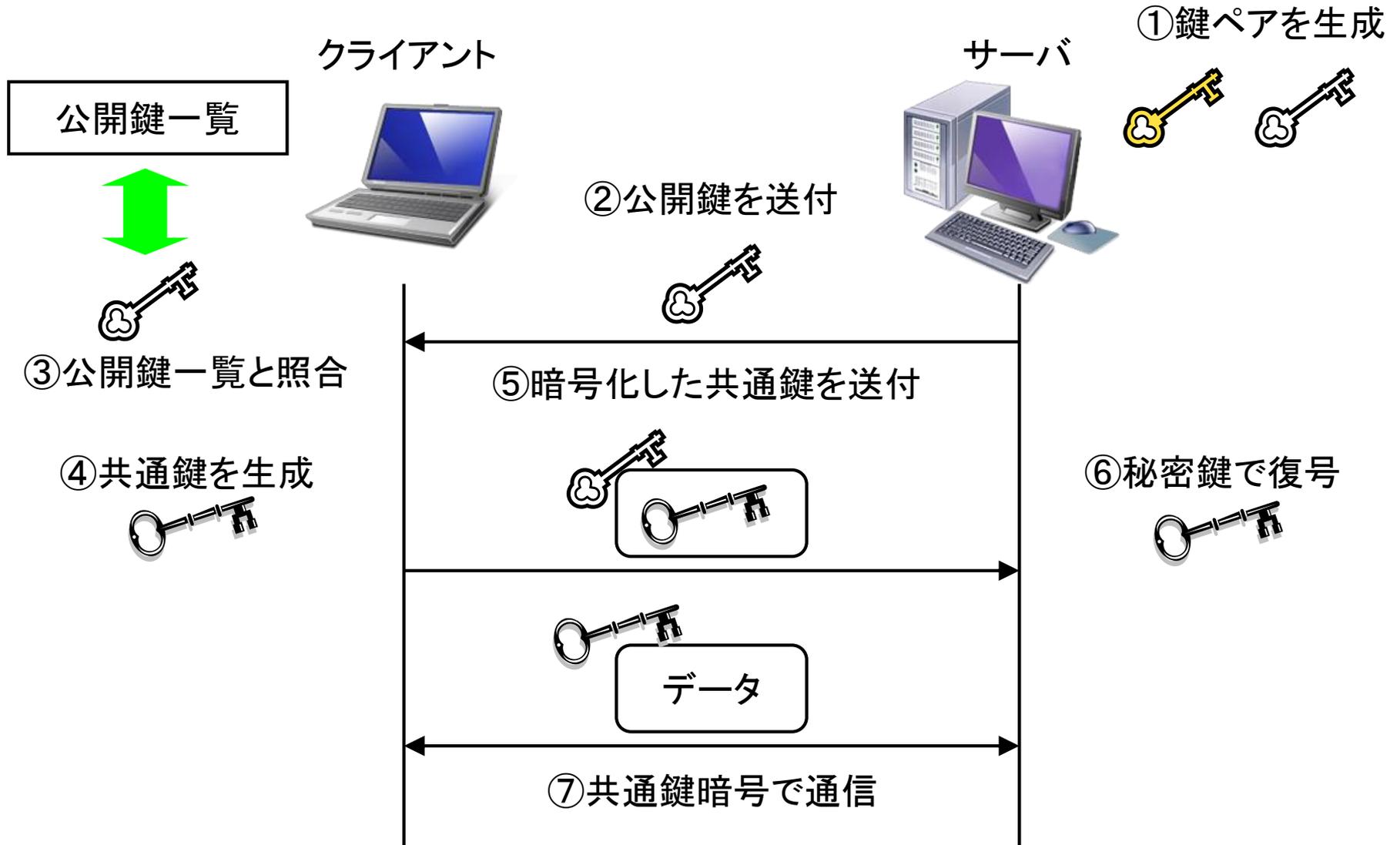
## ■ 認証技術が証明すること

- エンティティ(主体者)認証
  - メッセージの送信元(送信主体)に偽りが無いことを証明





# ホスト認証(サーバ認証)





## ■ホスト認証の手順

- ①鍵ペアはsshd起動時に自動生成
- ②サーバの公開鍵をクライアントへ送付
- ③送付された鍵が公開鍵一覧ファイルにあるかチェック  
(初回接続時はサーバ公開鍵を登録)

```
# ssh 192.168.1.100
The authenticity of host '127.0.0.1 (127.0.0.1)' can't be established.
RSA key fingerprint is
90:7f:9c:e7:c3:ad:89:75:c1:ae:9b:f0:20:e3:ad:89:75:c1:ae:9b:f0:20:e3:46:2d.
Are you sure you want to continue connecting (yes/no)?
```

### 公開鍵一覧ファイル名

- |                            |                 |
|----------------------------|-----------------|
| - /etc/ssh/ssh_known_hosts | システムレベルでのアクセス許可 |
| - ~/.ssh/ssh_known_hosts   | ユーザレベルでのアクセス許可  |

- ④クライアントが共通鍵を生成
- ⑤共通鍵をサーバから送付された公開鍵で暗号化して送付
- ⑥サーバの秘密鍵で復号して共通鍵を取り出す



## ■ 公開鍵認証の手順

① 鍵ペアを生成



クライアント



サーバ



② 公開鍵を送付



③ 公開鍵一覧に登録



公開鍵一覧

④ サーバにアクセス

⑤ クライアント公開鍵を要求

⑥ クライアント公開鍵を送付



⑦ 公開鍵一覧と照合



## ■ 公開鍵認証の手順

### ① ssh-keygenコマンド クライアント側で鍵ペアを生成

```

$ ssh-keygen -t dsa
Generating public/private dsa key pair.
Enter file in which to save the key (/home/testuser/.ssh/id_dsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/testuser/.ssh/id_dsa.
Your public key has been saved in /home/testuser/.ssh/id_dsa.pub.
The key fingerprint is:
78:0a:25:d1:70:21:f7:39:ec:69:89:eb:5a:e3:07:2b testuser@test.example.com

$ ls -l /home/testuser/.ssh/
合計 8
-rw----- 1 testuser testuser 736 7月 15 13:30 id_dsa
-rw-r--r-- 1 testuser testuser 611 7月 15 13:30 id_dsa.pub

```

	秘密鍵	公開鍵
SSHv1 RSA	identity	identity.pub
SSHv2 DSA	id_dsa	id_dsa.pub
SSHv2 RSA	id_rsa	id_rsa.pub



## ■ 認証手順詳細

- ②ユーザの公開鍵をサーバに送付  
(安全に送付するにはscpコマンドを使用するとよい)

```
$ scp ~/.ssh/id_dsa.pub 192.168.1.100:~/.ssh/authorized_keys
testuser@192.168.1.100's password:
id_rsa.pub                               100% 238      804.2KB/s   00:00
```

- ③サーバ側のユーザホームディレクトリに公開鍵一覧ファイル  
(~/.ssh/authorized\_keys)を作成し、①で生成したファイルを登録
- ④クライアントからssh接続
- ⑤クライアント公開鍵の要求
- ⑥クライアント公開鍵の送付

```
$ ssh 192.168.1.100
Enter passphrase for key `~/home/testuser/.ssh/id_dsa': パスワードを入力
Last login: Sat Jul 15 16:30:00 2011 from test.example.com
```



## ■ ssh-agent

- sshクライアント認証エージェント
- クライアント上でデーモンが起動
- メモリ上に秘密鍵を保持し、必要なときに利用

### ○ ssh-agentを利用し、秘密鍵を登録

```
$ ssh-agent bash          >> ssh-agentの子プロセスとしてbashを起動
$ ssh-add                 >> 秘密鍵を登録
Enter passphrase for /home/testuser/.ssh/id_dsa:
Identity added: /home/testuser/.ssh/id_dsa (/home/testuser/.ssh/id_dsa)
```

### ○ sshサーバへログイン

```
$ ssh 192.168.1.100
Last login: Sat Jul 15 16:30:00 2011 from test.example.com
```



## ■ /etc/ssh/sshd\_config

〈書式〉 キーワード 値

- 1行1エントリで記述
- 各項目はデフォルト値を持っており、記述がない場合は規定値を採用

```
PasswordAuthentication no
PermitRootLogin no
AllowUsers murata
PubKeyAuthentication yes
```



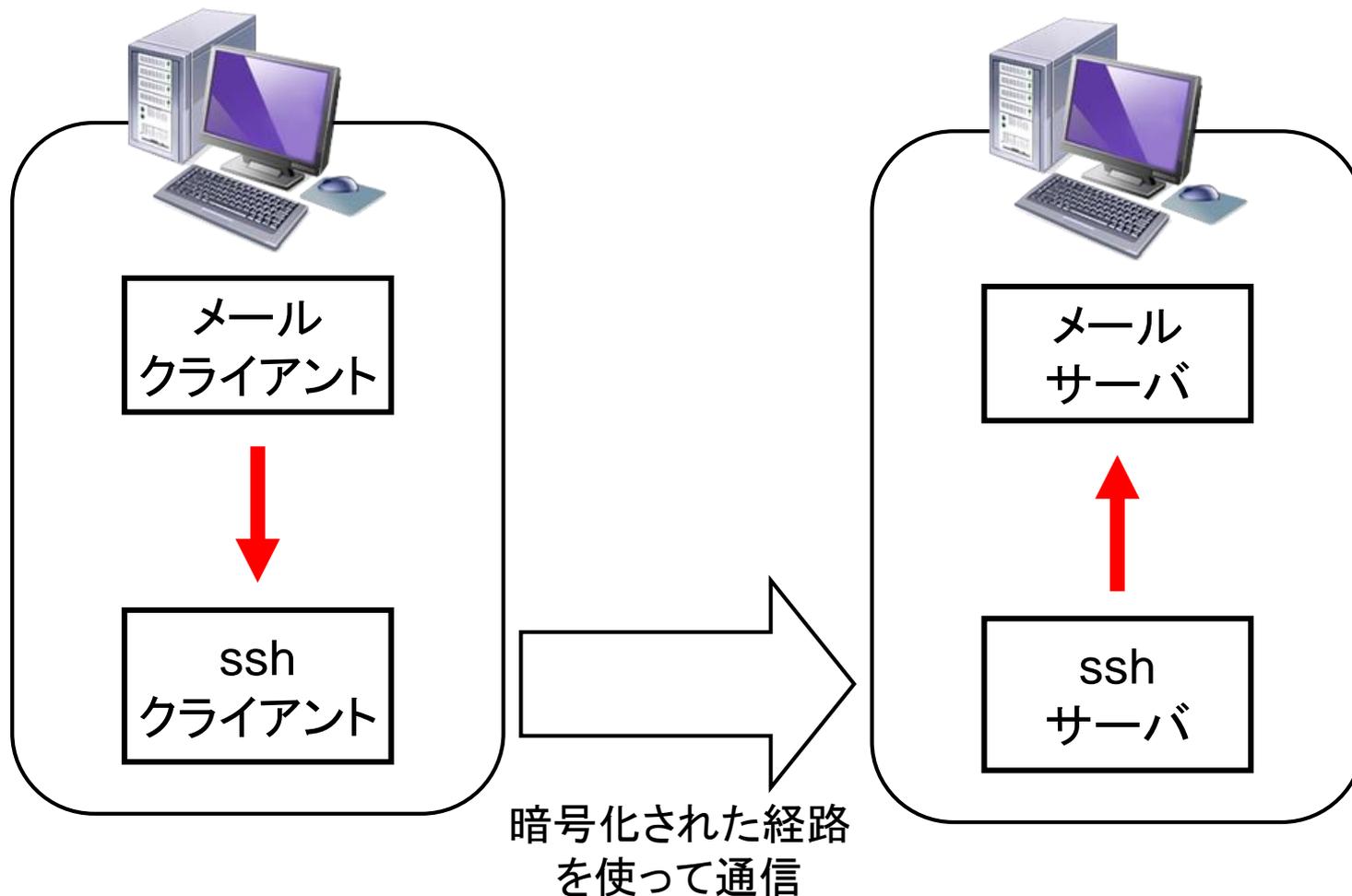
# SSH設定ファイル

## ○主なキーワード(設定項目)

PermitRootLogin	rootログインの許可設定
PermitEmptyPasswords	空パスワードの有効設定
PasswordAuthentication	パスワード認証の有効設定 (SSHv1, SSHv2)
RSAAuthentication	RSA公開鍵認証の許可設定 (SSHv1)
PubkeyAuthentication	公開鍵認証の許可設定 (SSHv2)
AuthorizedKeyFile	公開鍵を登録するファイル
X11Forwarding	X11フォワーディングの有効設定
UsePAM	PAMの使用設定
AllowUsers	接続を許可するユーザ
DenyUsers	接続を禁止するユーザ



# ポートフォワーディング





## ■ カスタマイズ研修のご案内

- LPIC試験対策研修
- Linux基礎、Linuxサーバ構築
- その他、ネットワーク・セキュリティ・XML・Web技術など  
各種IT研修をカスタマイズして提供

弊社研修サービスホームページ

<http://www.kcc.co.jp/product-service/it/>



株式会社 ケイシーシー

ご清聴ありがとうございました