

LPICレベル2技術解説セミナー

2013/3/16

NECラーニング株式会社
堀 邦彰

<http://www.neclearning.jp/>



■LPI201試験

- Linuxカーネル(10)
- システムの起動(8)
- ファイルシステムとデバイス(10)
- 高度なストレージ管理(6)
- ネットワーク構成(13)
- システムの保守(7)
- ドメインネームサーバ(6)

■LPI202試験

- Webサービス(6)
- ファイル共有(8)
- ネットワーククライアントの管理(7)
- 電子メールサービス(7)
- システムのセキュリティ(13)
- トラブルシューティング(19)

()は重要度合計 = 出題数



受験に向けて

- 重要度を考慮しながら、試験範囲をしっかりと覚えよう
- マシンで確認しましょう



- 重要度を考慮しながら、試験範囲をしっかりと押さえる

201.5 実行時におけるカーネルおよびカーネルモジュールの管理/照会

重要度	3
説明	2.6.xまたは3.xカーネルとそのロード可能なモジュールについての管理や照会ができる。
主要な知識範囲	<ul style="list-style-type: none">• <u>コマンドラインユーティリティを使用して、現在実行中のカーネルおよびカーネルモジュールに関する情報を取得する</u>• <u>手作業でカーネルモジュールをロードおよびアンロードする</u>• モジュールをアンロードできるタイミングを判断する• <u>モジュールが受け取るパラメータを判断する</u>• <u>モジュールをファイル名ではなく別の名前ロードできるようにシステムを設定する</u>
重要なファイル、用語、ユーティリティ	<ul style="list-style-type: none">• /lib/modules/kernel-version/modules.dep• /etc内のモジュール設定ファイル• /proc/sys/kernel/• depmod ・ insmod ・ lsmod ・ rmmod• modinfo ・ modprobe ・ uname



LPI201試験のポイント

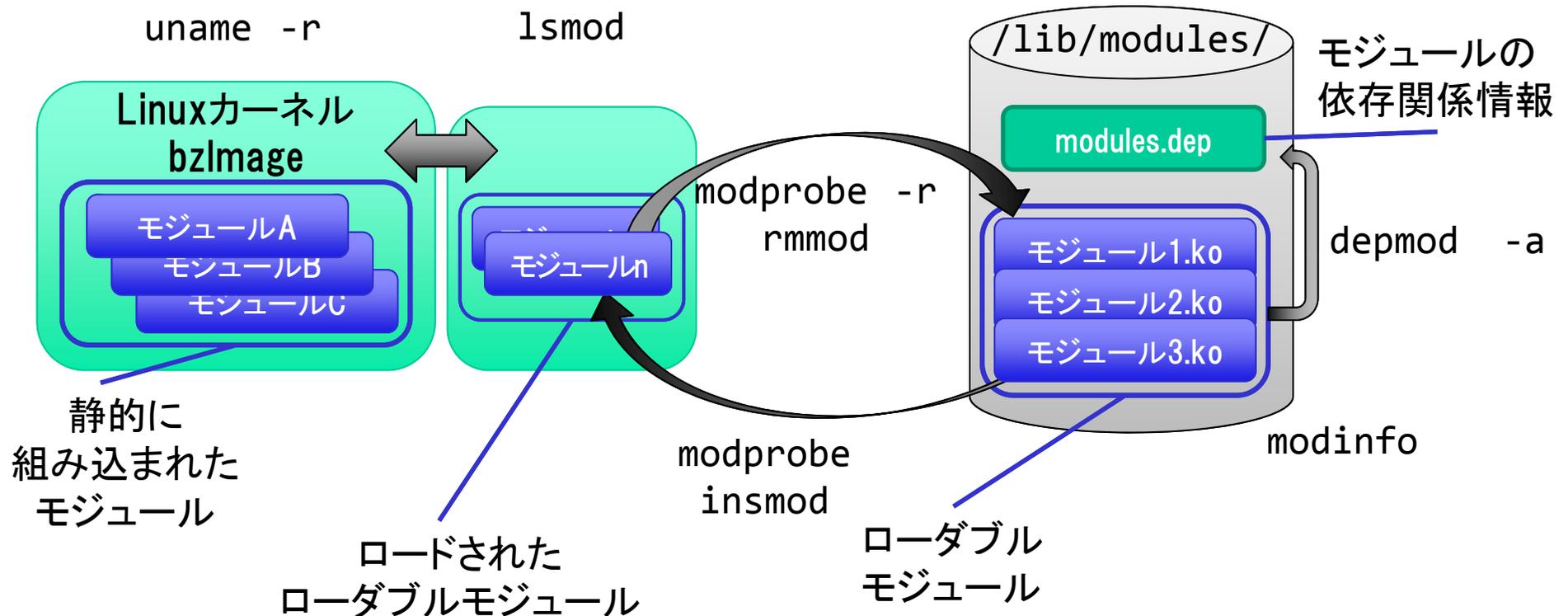
- Linuxカーネル(10)
- システムの起動(8)
- ファイルシステムとデバイス(10)
- 高度なストレージ管理(6)
- ネットワーク構成(13)
- システムの保守(7)
- ドメインネームサーバ(6)



- 201.1 カーネルの構成要素(2)
- 201.2 カーネルのコンパイル(2)
- 201.3 カーネルへのパッチ適用(1)
- 201.4 カスタムカーネルおよびカーネルモジュールのカスタマイズ、構築、インストール(2)
- 201.5 実行時におけるカーネルおよびカーネルモジュールの管理/照会(3)



- カーネルの各機能（SCSIのサポート、quota機能、パケットフィルタリング等）はモジュール化されており、カーネルモジュールと呼ばれる。
 - カーネルに静的に組み込む場合 → カーネルの再構築
 - 動的に呼び出す場合 → ロードやアンロードして使用





- カーネルの構築手順

	コマンドライン	作業項目
1	<pre># cd /usr/src/linux; vi Makefile</pre>	Makefile の編集(バージョンの設定)
2	<pre># make mrproper</pre>	カーネル設定の初期化
3	<pre># make *config (make xconfig、make oldconfig 等)</pre>	カーネル構成の設定
4	<pre># make または # make bzImage; make modules</pre>	カーネルイメージ、 ロードブルモジュールの作成
5	<pre># make modules_install</pre>	ロードブルモジュールの インストール
6	<pre># make install または # cp kernelImage /boot # mkinitrd /boot/initrd.img kernel-ver # vi /boot/grub/grub.conf</pre>	カーネルのインストール、 イニシャル RAM ディスクの作成、 ブートルoaderの設定

イニシャルRAMディスクの理解には
ブートプロセスの理解が必要！



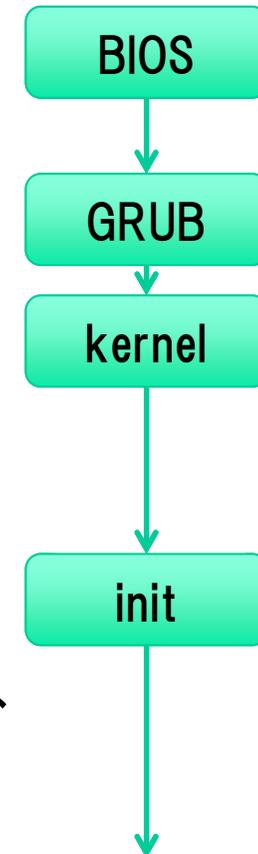
システムの起動(LPI201試験 2)

- 202.1 システムの起動とブートプロセスのカスタマイズ(4)
- 202.2 システムを回復する(4)



• ブートプロセス

1. BIOS 起動。ハードウェアの初期化とブートデバイスの決定を行う
2. BIOS がディスクの MBR から ブートローダー (GRUB) を読み出す
3. GRUB がカーネルとイニシャルRAMディスクを読み込む
4. カーネル起動
 - イニシャル RAM ディスクを暫定的なルートファイルシステムとしてマウント
 - 各種デバイスの初期化および実際のルートファイルシステムをマウントするために必要なカーネルモジュールをロード
 - 実際のルートファイルシステムをマウント
5. init プログラム (プロセスID: 1) を実行
6. init プログラムは `/etc/inittab` の設定に従い、ランレベルを決定、初期化スクリプト (`/etc/rc.d/rc.sysinit`) を実行
7. `/etc/rc.d/rc` スクリプトを実行し、各種サービスを起動
8. ログインプロンプトを表示





- /etc/inittab

- initの設定ファイル 書式 `id:runlevels:action:process`

```
id:5:initdefault:

si::sysinit:/etc/rc.d/rc.sysinit

l0:0:wait:/etc/rc.d/rc 0
l1:1:wait:/etc/rc.d/rc 1
...
l6:6:wait:/etc/rc.d/rc 6

ca::ctrlaltdel:/sbin/shutdown -t3 -r now

1:2345:respawn:/sbin/mingetty tty1
...
6:2345:respawn:/sbin/mingetty tty6

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

action	意味
initdefault	デフォルトのランレベルを決定する。 process の欄は無視される。
sysinit	システムのブート中に実行する。 runlevels の欄は無視される。
wait	指定したランレベルで processの プロセスを起動し、終了をまつ。
ctrlaltdel	[Ctrl]+[Alt]+[Delete]キーが押 された場合に実行される。
respawn	process に指定されたプロセスを起動 し、プロセスが終了した場合は常に再起 動する
once	指定したランレベルになった時に一度だ け実行する。
powerfail	電源に異常が起きた時に実行される。 電源異常は、UPS と通信して いるプロセスから知らされる。



- ブートプロセスのカスタマイズ
 - ブートローダーの設定を変更し、カーネル起動時にパラメータを与える
 - /sbin/init が実行するサービスの起動設定を変更

/boot/grub/grub.conf

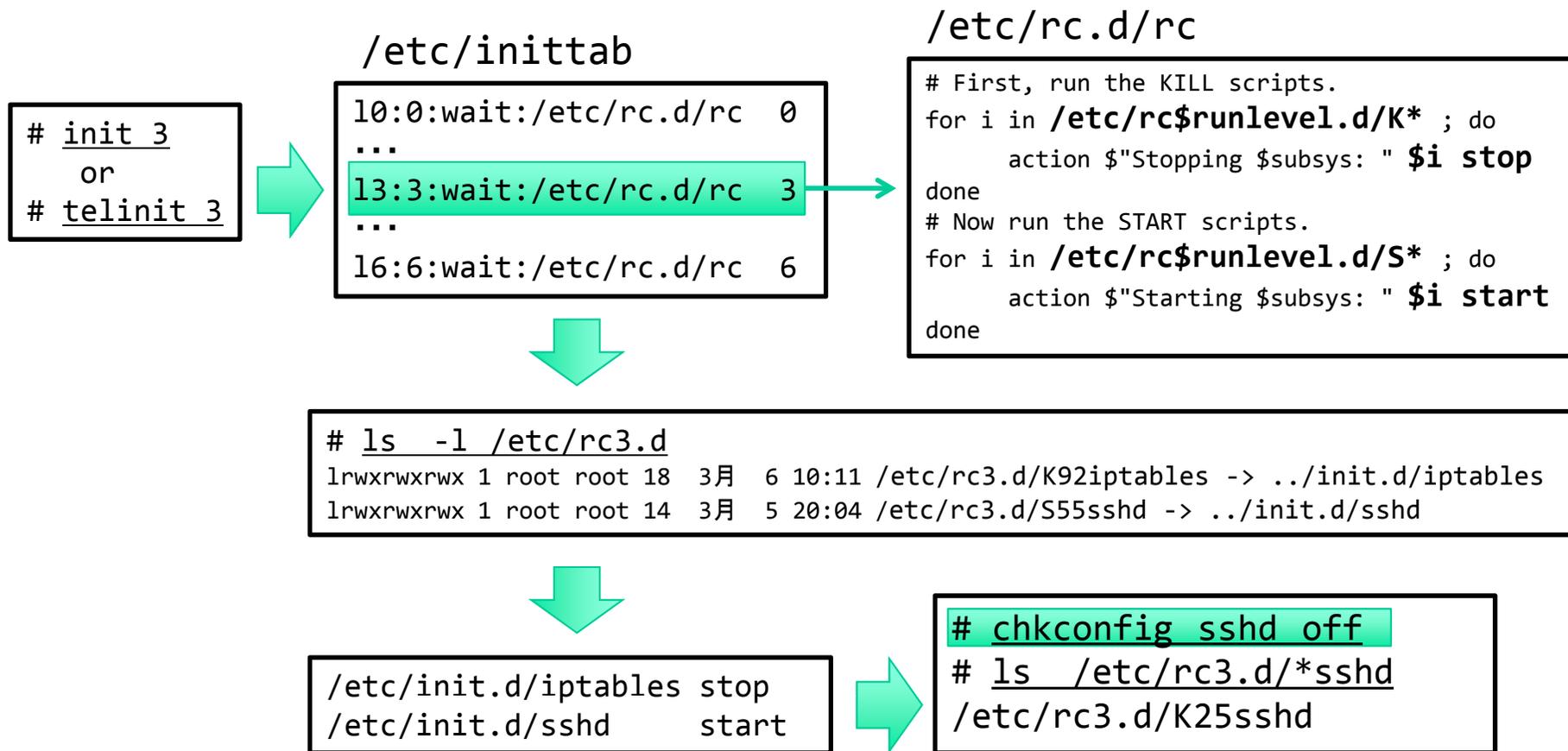
```
default=0
timeout=5
title Linux
  root (hd0,0)
  kernel /vmlinuz-2.6.9-5 ro root=LABEL=/
  initrd /initrd-2.6.9-5.img
title Linux Single
  root (hd0,0)
  kernel /vmlinuz-2.6.9-5 ro root=LABEL=/ S
  initrd /initrd-2.6.9-5.img
```

主なカーネル引数

記述	意味
root=ルートファイルシステム	システム起動時にマウントするルートファイルシステムを指定する。
ro、rw	ルートファイルシステムを読み取り専用でマウントを行う(ro)、または読み書き可能でマウントを行う(rw)
init=プログラム	カーネルの起動後に実行するプログラムを指定する。指定がない場合、init=/sbin/init となる。



- ブートプロセスのカスタマイズ
 - ブートローダーの設定を変更し、カーネル起動時にパラメータを与える
 - **/sbin/init が実行するサービスの起動設定を変更**





- 203.1 Linuxファイルシステムを操作する (4)
- 203.2 Linuxファイルシステムの保守 (3)
- 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 (2)
- 203.4 udevでのデバイス管理 (1)



- /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

デバイス	マウント ポイント	fsタイプ	マウント オプション	ダンプ	fsck
LABEL=/	/	ext3	defaults	1	1
LABEL=/boot	/boot	ext3	defaults	1	2
LABEL=/home	/home	ext3	defaults,ac1	1	2
LABEL=SWAP-hda6	swap	swap	defaults	0	0
/dev/hda8	/mydir	ext3	defaults,user	1	2

主なオプション	意味
defaults	rw、suid、dev、auto、nouser、asyncを使用する。
async/sync	すべての入出力を非同期/同期で行う。
(no)atime	アクセス時間を更新する(しない)。
(no)auto	自動マウントを行う(行わない)。
(no)user	一般ユーザーのマウントを許可(禁止)する。
ro/rw	ファイルシステムを読み取り専用(ro)、書き込み可能(rw)でマウントする。
usrquota	ユーザーquotaを使用する。
ac1	ACL (Access Control List)を使用する。



- 主なファイルシステム関連コマンド(ext2、ext3、ext4)

コマンド	用途
mke2fs	ext2/ext3/ext4 ファイルシステムを作成する -t ファイルシステムタイプの指定 -b ブロックサイズの指定 -m 予約済みブロックの指定 -j ジャーナルの付加
tune2fs	ext2/ext3/ext4 ファイルシステム内のパラメータを調整する -m 予約済みブロックの変更 -j ジャーナルの付加 -c 最大マウント回数の指定 -i ファイルシステムチェック間隔の指定 -l スーパーブロックの内容を表示
dumpe2fs	ext2/ext3/ext4 ファイルシステムの情報を表示する -h スーパーブロックの内容を表示
e2fsck	ext2/ext3/ext4 ファイルシステムをチェックする
debugfs	ext2/ext3/ext4 ファイルシステムのデバック用コマンド
e2label	ext2/ext3/ext4 ファイルシステムのラベルを設定する



- 204.1 RAIDを構成する (2)
- 204.2 記憶装置へのアクセス方法を調整する (1)
- 204.3 論理ボリュームマネージャ (3)

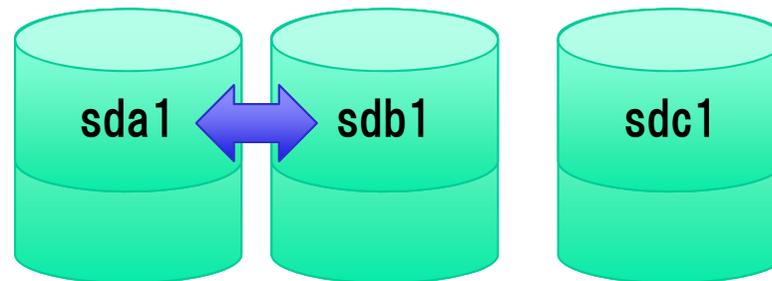


- ソフトウェアRAIDの管理はmdadmコマンドで行う。
- ソフトウェアRAIDの状況は /proc/mdstatで確認できる。

```
# mdadm --create /dev/md0 --level=1  
  --raid-devices=2 --spare-devices=1 /dev/sda1 /dev/sdb1 /dev/sdc1  
mdadm: array /dev/md0 started.
```

```
# cat /proc/mdstat  
Personalities : [raid1]  
md0 : active raid1 sdc1[2](S) sdb1[1] sda1[0]  
      104320 blocks [2/2] [UU]  
unused devices: <none>
```

```
# mdadm --stop /dev/md0  
# cat /proc/mdstat  
Personalities : [raid1]  
unused devices: <none>
```

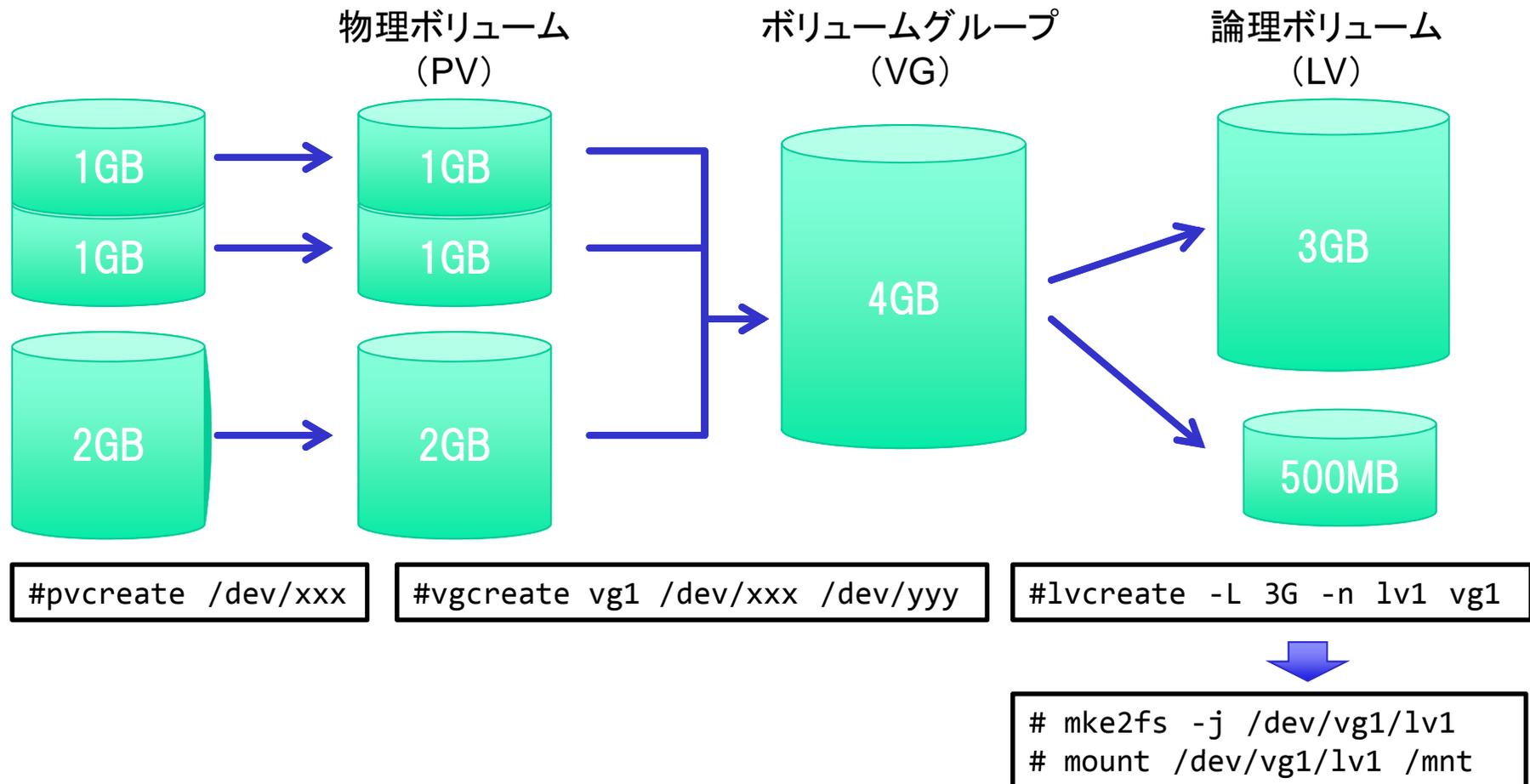


RAID-1(ミラーリング)

スペア



- 複数のディスクをまとめて1つの論理的なディスクとして取り扱う





ネットワーク構成 (LPI201試験 5)

- 205.1 基本的なネットワーク構成 (3)
- 205.2 高度なネットワーク構成とトラブルシューティング (4)
- 205.3 ネットワークの問題を解決する (5)
- 205.4 システム関連の問題をユーザに通知する (1)



- ifconfig ... IPアドレスの設定
 - ifconfig eth0 192.168.1.1 IPアドレスの設定
 - ifconfig eth0:X 192.168.1.100 IPエイリアスの設定
- route ... ルーティングテーブルの設定
 - route add -net 10.0.0.0/8 gw 192.168.1.100
 - route add -host 10.1.1.1 gw 192.168.1.250
 - route add default gw 192.168.1.254
- netstat ... ネットワークの状態確認
 - -i インターフェイスの状態テーブルの表示
 - -s 各プロトコルの統計情報の表示
 - -r ルーティングテーブルの表示



- `tcpdump` ... 条件に合致するネットワーク上のパケットのヘッダ表示

```
# tcpdump tcp port 23
tcpdump: listening on eth0
22:50:44.210803 192.168.1.101.46350 > 192.168.1.38.23: S 1694865396:1694865396(0) win
5840 <mss 1460,sackOK,timestamp 4175378 0,nop,wscale 2> (DF) [tos 0x10]
22:50:44.210878 192.168.1.38.23 > 192.168.1.101.46350: S 1849626719:1849626719(0) ack
1694865397 win 5792 <mss 1460,sackOK,timestamp 5104819 4175378,nop,wscale 0> (DF)
22:50:44.211042 192.168.1.101.46350 > 192.168.1.38.23: . ack 1 win 1460
<nop,nop,timestamp 4175378 5104819> (DF) [tos 0x10]
22:50:44.223408 192.168.1.38.23 > 192.168.1.101.46350: P 1:13(12) ack 1 win 5792
<nop,nop,timestamp 5104820 4175378> (DF) [tos 0x10]
```



- 206.1 ソースからプログラムをmakeしてインストールする (4)
- 206.2 バックアップ操作 (3)



1. アーカイブファイルの展開

tar zxvf xxx.tar.gz または # tar jxvf xxx.tar.bz2

2. ドキュメントの確認

- INSTALLやREADMEといったドキュメントを参照

3. ./configure の実行

- インストール環境のチェック等を行い、Makefileを作成する

4. make clean の実行

- 前回のコンパイル時の削除

5. make の実行

- ソフトウェアのコンパイル

6. make install の実行

- コンパイルしたプログラムのインストール。
所定の位置(/usr/local/binなど)にプログラムがコピーされるので、
rootユーザーで実行する



- 207.1 DNSサーバの基本的な設定 (2)
- 207.2 DNSゾーンの作成と保守 (2)
- 207.3 DNSサーバを保護する (2)



DNSサーバー(BIND)の設定

- 例) example.comドメイン(192.168.1ネットワーク)の管理

/etc/named.conf

```
options {  
    directory "/var/named";  
    version "x.x.x";  
};  
  
zone "." IN {  
    type hint;  
    file "named.ca";  
};  
  
zone "example.com" IN {  
    type master;  
    file "exam.db";  
    allow-update { none; };  
};  
  
zone "1.168.192.in-addr.arpa" IN {  
    type master;  
    file "1.168.192.db";  
    allow-update { none; };  
};
```

/var/named/exam.db

```
$TTL 86400  
example.com. IN SOA sv1.example.com. root.sv1.example.com. (  
    2005083101 28800 14400 602480 86400 )  
  
example.com.      IN      NS      sv1.example.com.  
example.com.      IN      MX      5      sv1.example.com.  
  
station1.example.com.  IN      A      192.168.1.1  
station2.example.com.  IN      A      192.168.1.2  
sv1.example.com.      IN      A      192.168.1.100  
www.example.com.      IN      CNAME     sv1.example.com.
```

/var/named/1.168.192.db

```
$TTL 86400  
1.168.192.in-addr.arpa. IN SOA sv1.example.com. root.sv1.example.com. (  
    2005083101 28800 14400 602480 86400 )  
1.168.192.in-addr.arpa.      IN      NS      server1.example.com.  
  
1.1.168.192.in-addr.arpa.    IN      PTR     station1.example.com.  
2.1.168.192.in-addr.arpa.    IN      PTR     station2.example.com.  
100.1.168.192.in-addr.arpa.  IN      PTR     sv1.example.com.
```

- クライアントは問い合わせるDNSサーバーを/etc/resolv.confに指定



■ DNS のセキュリティを高める技術

- DNSSEC

ゾーン用の鍵(公開鍵・秘密鍵)を利用して、
ゾーンデータに署名を行うことにより、ゾーンデータの検証を可能とする技術

- TSIG

ホスト用の鍵(共通秘密鍵)をプライマリサーバーとセカンダリサーバーが
互いに保存し、ゾーン転送などサーバー間の要求に署名(トランザクション署名)
をつける技術

■ どちらもdnssec-keygenコマンドで作成した鍵を利用



LPI202試験のポイント

- Webサービス(6)
- ファイル共有(8)
- ネットワーククライアントの管理(7)
- 電子メールサービス(7)
- システムのセキュリティ(13)
- トラブルシューティング(19)



Webサービス（LPI202試験 1）

- 208.1 Webサーバの実装（3）
- 208.2 Webサーバの保守（2）
- 208.3 プロキシサーバの実装（1）



- /etc/httpd/conf/httpd.conf
 - 主なディレクティブ

ディレクティブ	意味
Listen (Port)	サービス要求を受け付けるポート番号を指定。通常は 80 を指定。 (Listenの場合、インターフェイスとポート番号)
DocumentRoot	コンテンツ(HTML ファイル等)を格納するトップディレクトリを指定。 例) DocumentRoot "/var/www/html" 要求URL http://ホスト名/abc/xyz.html 配信ファイル /var/www/html/abc/xyz.html
UserDir	一般ユーザーが作成するコンテンツを格納するディレクトリを指定。 URLに「~ユーザー名」が指定された場合、そのユーザーのホームディレクトリ内のUserDirに指定したディレクトリにマッピングする。 例) UserDir public_html 要求URL http://ホスト名/~tom/contents/abc.html 配信ファイル /home/tom/public_html/contens/abc.html



- ユーザー認証

- AuthType, AuthName, AuthUserFile, Requireの組み合わせで設定
- htpasswdコマンドを使用して、パスワードファイルを作成

httpd.conf

```
<Directory "/var/www/html/secret">  
  AuthType      Basic  
  AuthName      "Member's Only!"  
  AuthUserFile  /etc/httpd/.htpasswd  
  Require       valid-user  
</Directory>
```



```
# htpasswd -cm /etc/httpd/.htpasswd taro  
New password: パスワードを設定  
Re-type new password: 設定したパスワードを再入力  
Adding password for user taro
```

ディレクティブ	意味
AuthType	認証方法の指定。引数は、"Basic", "Digest"のどちらか。 Basic ... ユーザー名、パスワードを平文で送信。 Digest ... ユーザー名、パスワードを MD5 で暗号化して送信。
AuthName	ユーザー認証のダイアログに表示される文字列を指定
AuthUserFile	ユーザー認証に使用するパスワードファイル(.htpasswdなど)を指定。
Require	正しく認証されたユーザーの中でアクセスを認めるユーザーの指定。 valid-user 正しく認証されたユーザーならば、全てアクセスを許可。 user username [username ...] 正しく認証され、かつusername に指定されたユーザーのみアクセス許可。



ファイル共有（LPI202試験 2）

- 209.1 Sambaサーバの設定（4）
- 209.2 NFSサーバの設定（4）



- /etc/samba/smb.conf

```
[global]
workgroup = LPI
server string = Samba Server Version %v
security = user
passdb backend = smbpasswd:/etc/samba/smbpasswd
map to guest = Bad password
username map = /etc/samba/smbusers
unix password sync = Yes
```

[global]セクション
Samba の全体設定や他の
セクションのパラメータの
デフォルト値の設定する

```
[homes]
comment = Home Directories
browseable = no
writable = yes
```

[homes]セクション
ユーザーのホームディレクトリを共有する。

```
[share]
comment = Shared Space
path = /share
writeable = Yes
guest ok = Yes
```

独自の共有セクションの例。
[]で指定したセクション名が共有名となる。
[共有名]
path = 共有先

- testparmコマンドで設定確認、smbclientコマンドで接続確認。



- NFSサーバーが公開(エクスポート)したディレクトリをNFSクライアントがマウントすることで、NFSサーバー上のファイルシステムをあたかもローカルシステム上のファイルシステムのように扱うことができる

/etc/exports

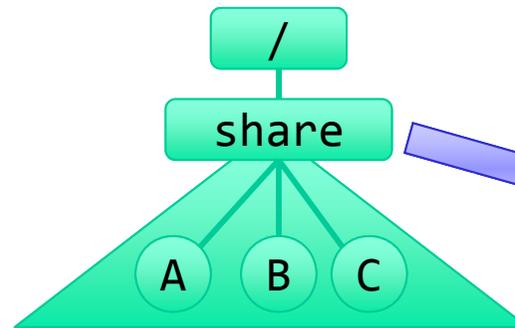
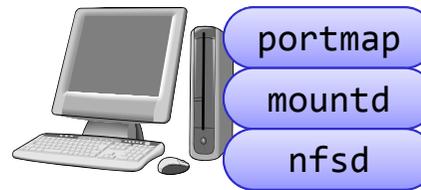
```
/share client1(rw, sync)
```



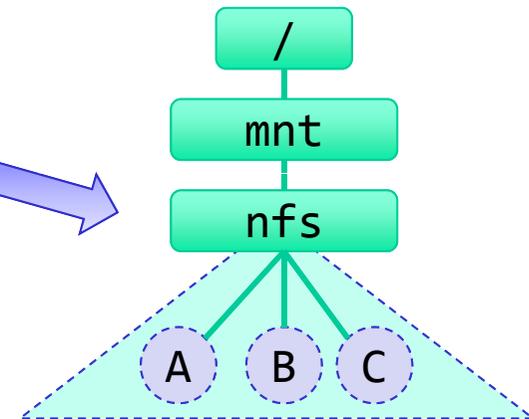
```
#exportfs -r
```

エクスポートオプション	意味
ro	読み取り専用(ro)でエクスポート
rw	書き込み可能(rw)でエクスポート
root_squash	クライアントのrootアクセスをanonymousアクセスとする
no_root_squash	クライアントのrootアクセスをrootアクセスとして認める
anonuid=uid	anonymousユーザーIDの定義
sync	同期動作

NFSサーバー
server1



クライアント
client1



```
#showmount -e server1
#mount -t nfs server1:/share /mnt/nfs
```



- 210.1 DHCPの設定 (2)
- 210.2 PAM認証 (3)
- 210.3 LDAPクライアントの利用方法 (2)



- サーバーはクライアントの要求に応じてIP アドレス情報などを貸し出す
- クライアントは、貸し出されるIPアドレス情報を用いてネットワークに接続

/etc/dhcpd.conf

```
option subnet-mask      255.255.255.0;
option broadcast-address 192.168.1.255;
option routers          192.168.1.254;
option domain-name     "example.com";
option domain-name-servers 192.168.1.1;
default-lease-time     21600;
max-lease-time         43200;

subnet 192.168.1.0 netmask 255.255.255.0 {

    range 192.168.1.100 192.168.1.149;

    host station150 {
        hardware ethernet 00:00:4C:12:34:CD;
        fixed-address 192.168.1.150;
        option domain-name "sub.example.com";
    }
}
```

全体設定

subnet
固有の設定

ホスト
固有の設定

/var/lib/dhcpd/dhcpd.lease

```
...
lease 192.168.1.149 {
    starts 2 2012/09/19 09:02:21;
    ends 2 2012/09/19 15:02:21;
    binding state active;
    next binding state free;
    hardware ethernet 00:00:4c:a3:5f:40;
}
...
動的に貸し出した IP アドレス情報
```

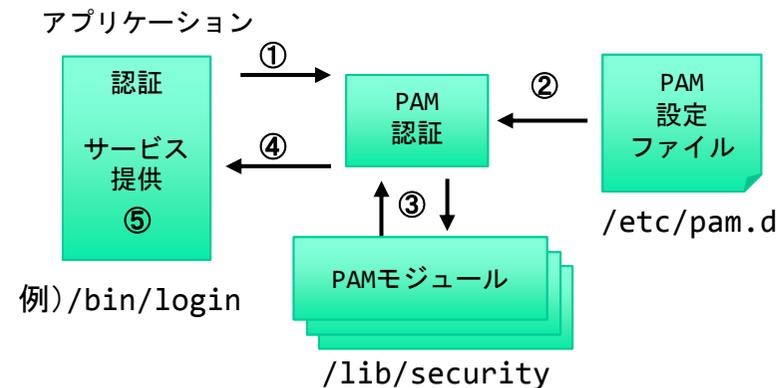


PAM認証

- 様々な認証方式をモジュールとしてアプリケーションに提供する仕組み
- 各アプリケーションは、PAMの利用により柔軟に認証手続きを変更可能

PAMを用いた認証

1. アプリケーションが認証を行なう際にPAMを呼び出す。
2. 各アプリケーション用のPAM設定ファイルを読み込む。
3. PAMモジュールが呼び出され認証が行われる。
4. 認証の結果がアプリケーションに返される。
5. サービスの提供が行われる。



/etc/pam.d/login

タイプ	コントロール	モジュール
auth	required	pam_securetty.so
auth	include	system-auth
account	required	pam_nologin.so
account	include	system-auth
password	include	system-auth
session	include	system-auth
session	optional	pam_console.so

タイプ	意味
auth	ユーザーの本人確認を行う。パスワードの確認など
account	サービスへのアクセスを許可されているかどうかを確認時刻やアクセス場所、パスワード期限など
password	ユーザーの認証方法を変更する。
session	サービスの開始・終了時に処理を行う

コントロール	意味
required	モジュールによる認証に失敗した場合に、全体の認証を拒否。ただし、他の同タイプのモジュールも実行
requisite	モジュールによる認証に失敗した場合に、即座に全体の認証を拒否(以降のモジュールは実行しない)。
sufficient	このモジュールによる認証が成功した場合、即座に全体の認証を許可(以降のモジュールは実行しない)。ただし、それまでのrequired型のモジュールがすべて成功している必要あり。失敗した場合には無視。
optional	認証の許可/拒否に関係しない動作を付加。
include	別のPAMの設定ファイルの読み込み、評価を行う。



- 211.1 電子メールサーバの使用（3）
- 211.2 ローカルの電子メール配信を管理する（2）
- 211.3 リモートの電子メール配信を管理する（2）



メールシステムの構成

- メール配送には多くの構成要素が関係

- MTA (Mail Transfer Agent)
メールの配送を担当するプログラム。
受け取ったメールは宛先に応じて他のMTAに配送したり、MDAに渡したりする。(sendmail、Postfix、qmailなど)

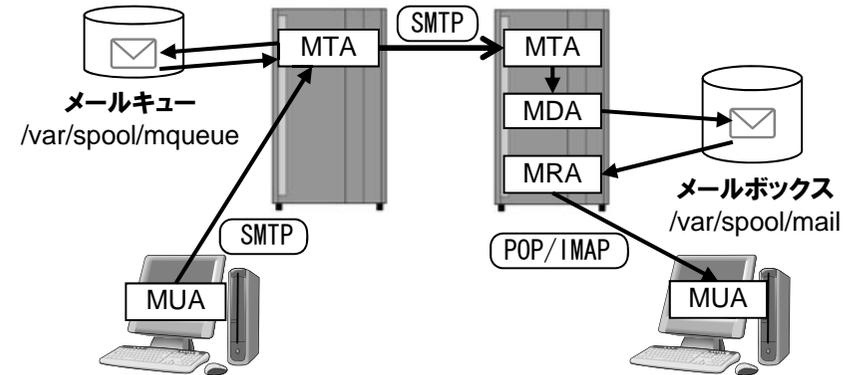
```
/etc/postfix/main.cf
```

```
mydestination = $myhostname, $mydomain
```

- MDA (Mail Delivery Agent)
MTAからメールを受け取り、メールボックスに格納する。
(procmailなど)

```
/etc/procmailrc または ~/.procmailrc
```

```
:0 Bc  
* < 1048576  
* ! ¥<html  
! mickey@mydomain.com
```



- MRA (Mail Retrieval Agent)
メールボックス上のメールデータをMUAに渡す。
POPやIMAP通信によりメールを渡します。
(Dovecot、Curier-IMAPなど)

```
/etc/dovecot.conf
```

```
protocols = imap imaps  
auth default {  
  mechanisms = plain
```

- MUA (Mail User Agent)
メールの作成、表示をするクライアント側のプログラム。
送信時、サーバー上のMTAにメールを渡す。
受信時、サーバー上のMRAに接続し、メールを受信。
(Outlook、Thunderbirdなど)

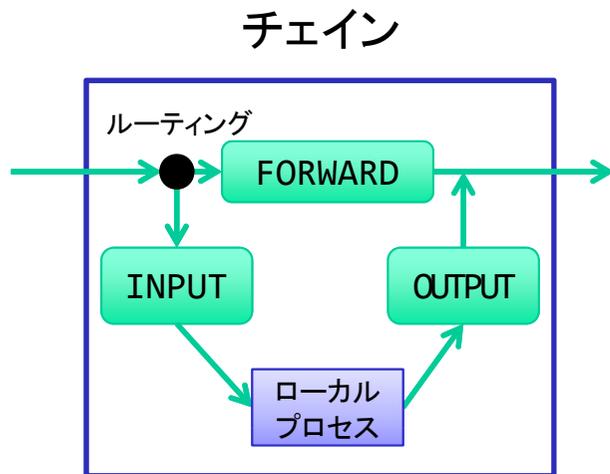


- 212.1 ルータを構成する (3)
- 212.2 FTPサーバの保護 (2)
- 212.3 セキュアシェル(SSH) (4)
- 212.4 TCPラッパー (1)
- 212.5 セキュリティ業務 (3)



- カーネルレベルでパケットフィルタリングを行う

```
iptables -A チェイン名 -s 送信元アドレス -d 送信先アドレス -j ターゲット
```



ルールオプション	意味
-s 送信元アドレス	送信元アドレスの指定
-d 送信先アドレス	送信先アドレスの指定
-p プロトコル [--sport ポート番号] [--dport ポート番号]	プロトコルを指定 --dportや--sportで対象パケットの 送信先および送信元ポート番号を指定。
-i インタフェース名	受信インタフェースの指定
-o インタフェース名	送信インタフェースの指定
-j ターゲット	該当パケットの扱い方の指定 ACCEPT 通過許可 DROP パケット破棄 REJECT パケット破棄(エラー通知)

```
192.168.1.0/24ネットワークからTELNET(23/tcp)接続を許可します。
#iptables -A INPUT -s 192.168.1.0/24 -p tcp --dport 23 -j ACCEPT
192.168.1.100 へのDNS(53/udp)接続を許可します。
#iptables -A OUTPUT -d 192.168.1.100 -p udp --dport 53 -j ACCEPT
192.168.1.0/24ネットワーク以外からのUDP接続は拒否する。
#iptables -A INPUT -s ! 192.168.1.0/24 -p udp -j REJECT
```



- /etc/ssh/sshd_config

```
Port 22
Protocol 2,1
PermitRootLogin yes
PubkeyAuthentication yes
HostbasedAuthentication yes
PasswordAuthentication yes
PermitEmptyPasswords no
UsePAM yes
X11Forwarding yes
```

ユーザー認証
方法の選択

PubkeyAuthentication yes

- ホスト制限: TCP wrapperを利用

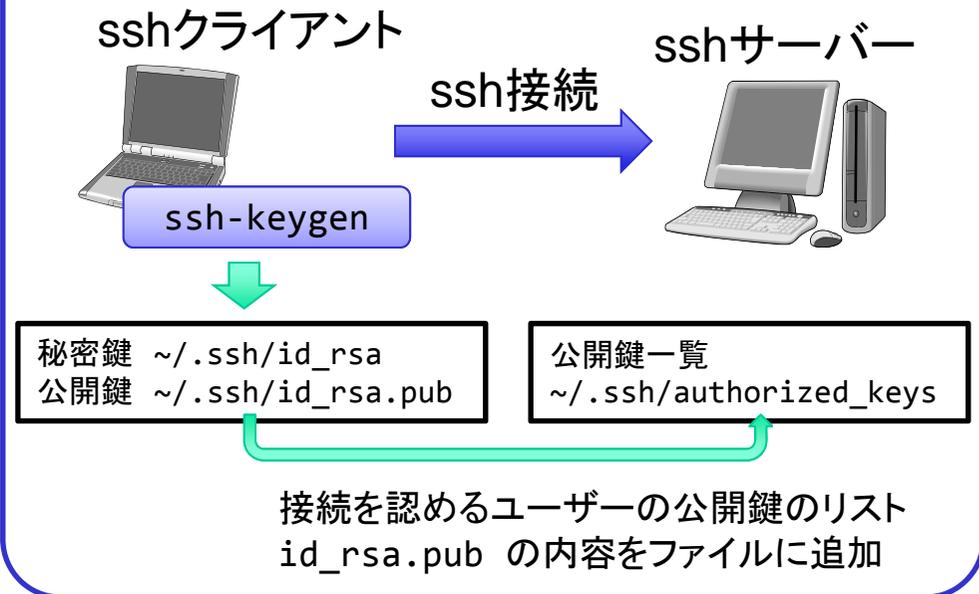
/etc/hosts.deny

```
sshd : ALL
```

/etc/hosts.allow

```
sshd : 192.168.1.0/255.255.255.0
```

ユーザー認証に公開鍵認証を使用するには





- 213.1 ブート段階の識別とブートローダのトラブルシューティング (4)
- 213.2 一般的な問題を解決する (5)
- 213.3 システムリソースの問題を解決する (5)
- 213.4 環境設定の問題を解決する (5)



- ブート段階の識別は、実際に確認してみるのが良い。

Linuxのブートプロセス

BIOS の起動

- ブートデバイスの決定
- ブートローダーの起動

ブートローダーの起動

- カーネルの決定、カーネルのロード

カーネルの起動

- ハードウェアの検出
- ルートファイルシステムのマウント(読み取り専用)

/sbin/init の実行①

- /etc/rc.d/rc.sysinit の実行
(/etc/fstab を参照し、以下の事柄が行われる)
 - (ア) ルートファイルシステムのチェック
 - (イ) ルートファイルシステムの再マウント(読み書き可)
 - (ウ) 他のファイルシステムのチェックおよびマウント

/sbin/init の実行②

- /etc/rc.d/rc の実行
/etc/rcX.d 内のスクリプト実行によるサービス起動
- mingetty などの起動によるログインプロンプトの表示
- X Window System の起動(ランレベル 5 の場合)

ログイン可能

ブートローダーの設定を
色々変えて起動してみる。

/boot/grub/grub.conf

```
default=0
timeout=5
title Linux
    root (hd0,0)
    kernel /vmlinuz-2.6.9-5 ro root=LABEL=/
    initrd /initrd-2.6.9-5.img
```



ご参考

■ NECラーニング

<http://www.neclearning.jp/>

NECラーニング Linuxトレーニング

ウェブ 画像 地図 ショッピング もっと見る ▾ 検索ツール

約 2,420,000 件 (0.23 秒)

[Linuxトレーニング/研修 | NECラーニング株式会社](#)
[www.neclearning.jp](#) > 研修サービス > IT研修 - キャッシュ
 linuxの基礎からサーバー構築・設定まで目的に応じたトレーニング/研修で、短期間で効率よく知識と技術を修得しましょう。

[OSS関連研修/トレーニング | NECラーニング株式会社](#)
[www.neclearning.jp](#) > 研修サービス > IT研修 - キャッシュ
 60件のアイテム - OSS関連研修/トレーニング. OSS(オープンソースソフトウェア)は、...

コース名	受講料(税込)
UNIX/Linux基礎1 - 基本機能とコマンド	78,750円
UNIX/Linux基礎2 - sed/awk/シェルスクリプト	63,000円

[LPIトレーニング/研修 | NECラーニング株式会社](#)
[www.neclearning.jp](#) > 研修サービス > IT研修 - キャッシュ
 LPI試験対策コースは、世界最大のlinux技術者認定試験、LPI資格取得を目指す方向向けのトレーニング/研修体系です。... NECラーニングでは、LPI資格取得を目指す方を対象に、試験対策トレーニング/研修を取り揃えています。LPI試験対策コースは、各試験に...