



LPI-Japan主催
LPIC レベル2
技術解説無料セミナー

株式会社びぎねっと

宮原 徹

tmiyahar@Begi.net

本日のゴール

1. LPIC Level2は出題範囲が広い
2. とても3時間では全部解説できない
3. 典型的なトピックを取り上げて、勉強方法を検討する
4. 例題を解いてみて、解き方を考える
5. あとは自分で頑張って勉強する
6. 見事合格！
7. 次はレベル3合格を目指す



本日のアジェンダ

1. 出題範囲の確認と学習対策
2. ポイント解説
 - I. Linuxカーネル
 - II. システムの起動
 - III. ファイルシステム
 - IV. セキュリティ
3. 例題を解いてみる



新出題範囲 (v3.0) について

- 2009年4月頃からリリース予定
- 主題が全体的にすっきりした
 - ニュース (NetNews) が無くなった！
 - PCMCIAが無くなった！
 - LVMがトピックとして独立
 - トラブルシューティングが1つにまとめられた



201試験の出題範囲(v3.0)

主題201:Linuxカーネル

| | |
|---|---|
| 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 |

主題202:システムの起動

| | |
|------------------------------|---|
| 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 |

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 | 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 |

主題204:高度なストレージ管理

| | |
|-------------------------|---|
| 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ー | 3 |

主題205:ネットワーク構成

| | |
|-------------------------------|---|
| 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ーに通知する | 1 |

主題206:システムの保守

| | |
|---------------------------------|---|
| 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 |

主題207:ドメインネームサーバー

| | |
|----------------------|---|
| 207.1 DNSサーバーの基本的な設定 | 2 |
| 207.2 DNSゾーンの作成と保守 | 2 |
| 207.3 DNSサーバーを保護する | 2 |



202試験の出題範囲(v3.0)

主題208:Webサービス

| | |
|-------------------|---|
| 208.1 Webサーバーの実装 | 3 |
| 208.2 Webサーバーの保守 | 2 |
| 208.3 プロキシサーバーの実装 | 1 |

主題209:ファイルとサービスの共有

| | |
|--------------------|---|
| 209.1 Sambaサーバーの設定 | 4 |
| 209.2 NFSサーバーの設定 | 4 |

主題210:ネットワーククライアントの管理

| | |
|-----------------------|---|
| 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 |

主題211:電子メールサービス

| | |
|-------------------------|---|
| 211.1 電子メールサーバーの使用 | 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 |

主題212:システムのセキュリティ

| | |
|--------------------|---|
| 212.1 ルーターを構成する | 3 |
| 212.2 FTPサーバーの保護 | 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 |

主題213:トラブルシューティング

| | |
|--|---|
| 213.1 ブート段階の識別と ブートローダー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 |



全体的な対策プラン(v3.0)

サーバ構築でカバーされる範囲

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205: ネットワーク構成
主題207: ドメインネームサーバー
主題208: Webサービス
主題209: ファイルとサービスの共有
主題211: 電子メールサービス

じっくりと

カバーされない範囲

主題201: Linuxカーネル
主題204: 高度なストレージ管理
主題206: システムの保守
主題210: ネットワーククライアントの
管理
主題212: システムのセキュリティ
主題213: トラブルシューティング

しっかりと



学習の方法

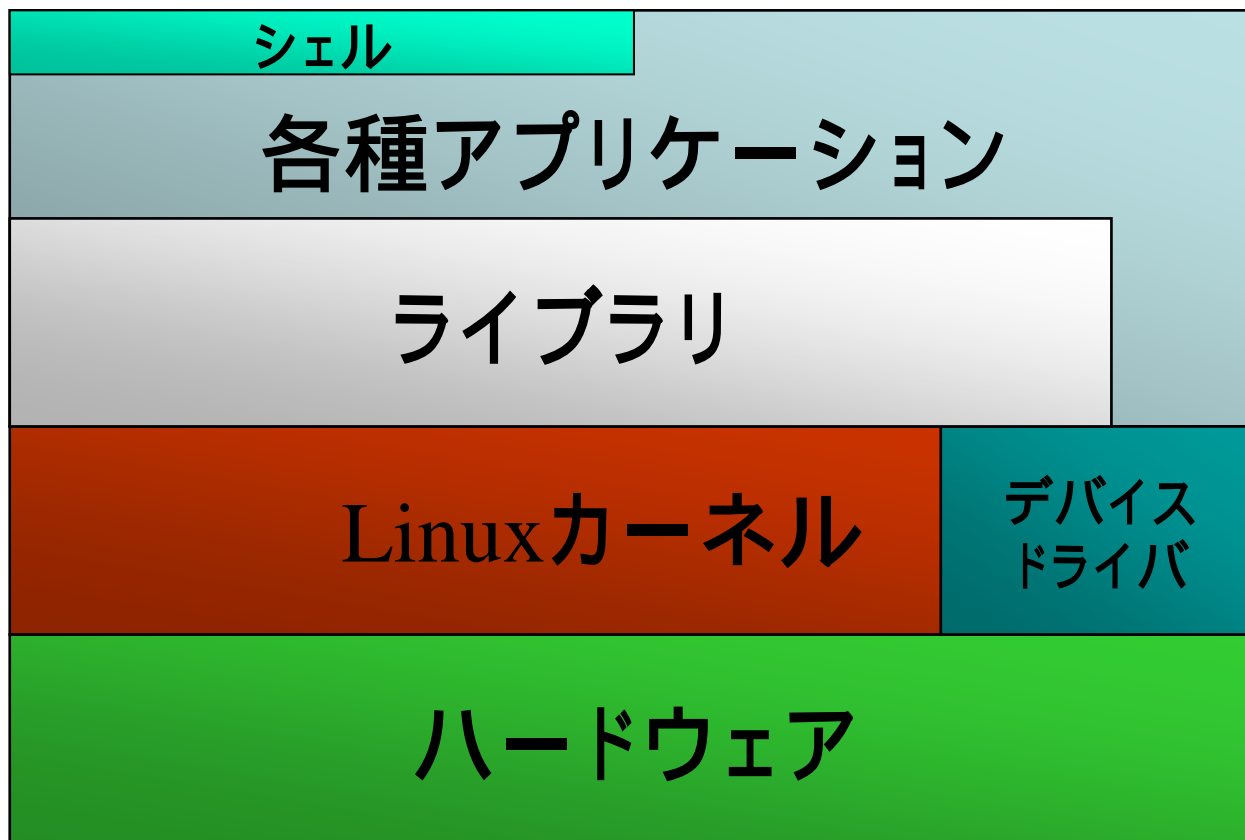
- 出題範囲をしっかりと把握
 - 関連キーワードはすべて調べる
- 一般的なネットワークサーバ構築を学習
 - IPアドレス、DNS、Web、メール、ファイルサーバ等
- セキュリティについて学習
 - TCP/IPネットワークの知識を再確認すること
- トラブルシューティングの技法
 - 基礎知識の再確認
 - 自分なりによくあるトラブルをまとめてみる



ポイント解説 カーネル



Linuxの構造



ユーザー
空間

カーネル
空間



カーネル再構築 (2.6系)

1. /usr/src/linuxにソースコードを展開 (tar, gzip, bzip2等)
2. 必要に応じてパッチを当てる (`patch -p0 < patch_file`)
3. (make mrproper 完全な消去)
4. make config カーネル設定
5. make clean 不要なものの消去
6. make **カーネルとモジュールの構築**
7. make modules_install モジュールインストール
8. make install カーネルのインストールと
ブートローダーの

設定



カーネル再構築 (2.4系)

1. /usr/src/linuxにソースコードを展開 (tar, gzip, bzip2等)
2. 必要に応じてパッチを当てる (`patch -p0 < patch_file`)
3. (make mrproper 完全な消去)
4. make config カーネル設定
5. make dep 依存関係チェック
6. make clean 不要なものの消去
7. make bzImage カーネルの構築
8. make modules モジュールの構築
9. make modules_install モジュールインストール
10. make install
トロードラーの設定



カーネル設定

- カーネルの機能をON/OFF/モジュール化
 - ONにした機能はカーネル本体に組み込まれ、起動時にメモリにロードされる
 - モジュールは動的にロードされる
- 設定方法はconfig / menuconfig / xconfig
 - 2.6ではdefconfig / allmodconfig / allyesconfig / allnoconfigなども使用可能
- 設定ファイルは.configファイル
 - 以前のバージョンの.configを再利用したい場合にはmake oldconfigを実行



モジュール関連

- 関連するファイル、ディレクトリ
 - インストール先: /lib/modules/[カーネルバージョン](#)/
 - 設定ファイル: /etc/modules.conf・[modprobe.conf](#)
- 関連するコマンド
 - depmod 依存関係の調査
 - lsmod ロードされているモジュールの一覧
 - modprobe 依存関係を解消してロード
 - insmod/rmmod モジュールのロード/アンロード
 - mkinitrd RAMディスク作成



ポイント解説 システムの起動



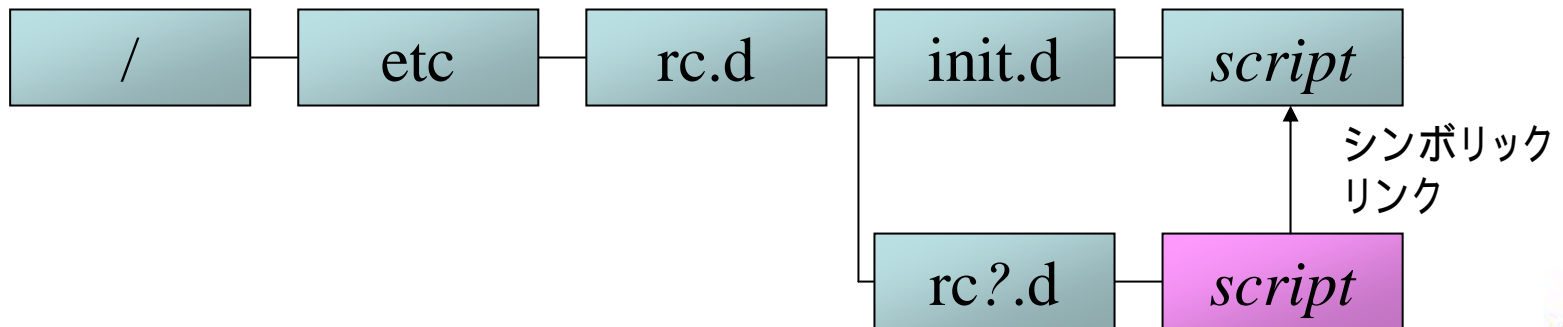
システム起動の順序

1. 電源ON
2. BIOSがPOST (Power On Self Test) を実行
3. 起動デバイスを決定
4. 起動デバイスのMBR (Master Boot Record・パーティションテーブルとIPL)を読み込み
5. ブートローダー (LILO・GRUBなど) を起動
6. カーネルを読み込み・起動
7. (RAMディスクを読み込み、モジュールをロード)
8. /(ルート)パーティションをマウント
9. initプロセスを起動 (/etc/inittabの指示に従う)



Linuxのサービス起動の仕組み

- /etc/rc.d/ディレクトリに関連ファイル
 - /etc/rc.d/init.d/ 起動スクリプトを格納
 - /etc/rc.d/rc?.d/ 各Runレベルでの起動スクリプトへのシンボリックリンクを格納



ポイント解説 ファイルシステム



Linuxのファイルシステム

- ext2
 - Linuxで標準的に使用されているファイルシステム
 - iノードでの管理を行う
- ext3
 - ext2にジャーナリング機能を追加
 - ext2と互換性あり
 - tune2fsコマンドで変換可能
- ReiserFS / XFS / JFS
 - ジャーナリング機能をサポートした・FS
 - 別途モジュールとファイルシステムの初期化が必要



ファイルシステム関連コマンド

- mount/umount
- sync
- swapon/swapoff
- fsck
- badblocks
- mke2fs/dumpe2fs/debuge2fs/tune2fs
- mkisofs
- dd



fstabの書式

ファイルシステム マウントポイント FSタイプ オプション ダンプ fsck

- ファイルシステム: デバイスファイル
 - e2labelコマンドでつけたラベル名の場合もある
- マウントポイント: ディレクトリ名
- FSタイプ: ファイルシステムの種類
 - ext2/ext3/iso9660/auto など
- オプション: mountコマンドのオプション
- dump: dumpコマンドの対象とするか
- fsck: fsckコマンドの対象・順序
 - /パーティションは必ず1、その他は2にする



マウントオプション

- `async/sync` 非同期/同期書き込み
- `atime/noatime` アクセス時間
- `auto/noauto` `mount -a`の対象にするか
- `dev/nodev` デバイスファイル作成
- `exec/noexec` ファイルの実行
- `suid/nosuid` `suid`ファイルの有効
- `rw/ro` 読み書き/読み取り専用
- `user/users/nouser` 一般ユーザもマウントできる/誰でもアンマウントできる/管理者のみ
- `defaults` `async,auto,dev,exec,nouser,suid,rw`
- `remount` 再度マウントし直す



ポイント解説 セキュリティ



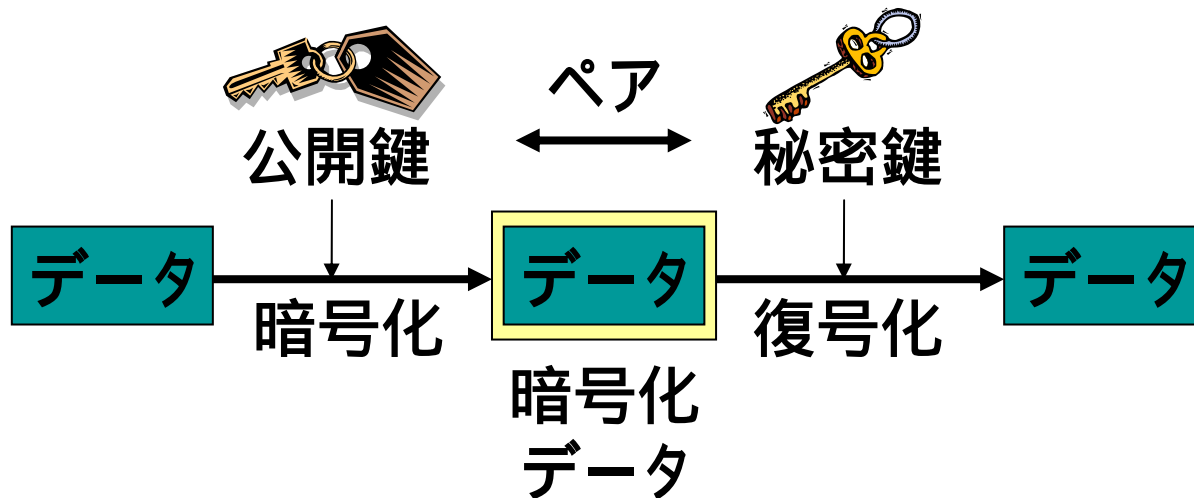
SSHによる通信の暗号化

- 通信の暗号化を行うセキュアなリモートシェル
 - telnet, rlogin, rsh, rcpなどと置き換えが可能
 - コマンドラインターミナルとしての利用
 - 各種認証が可能
 - ホストベース認証/パスワード認証/公開鍵認証/S/Key認証 (One Time Password認証)
 - RSA暗号(SSH v1)、DSA暗号(SSH v2)による公開鍵・秘密鍵暗号が利用可能
- LinuxではOpenSSHを使用
 - OpenSSHはOpenBSD開発グループが開発を行っているフリーなSSHソフトウェア

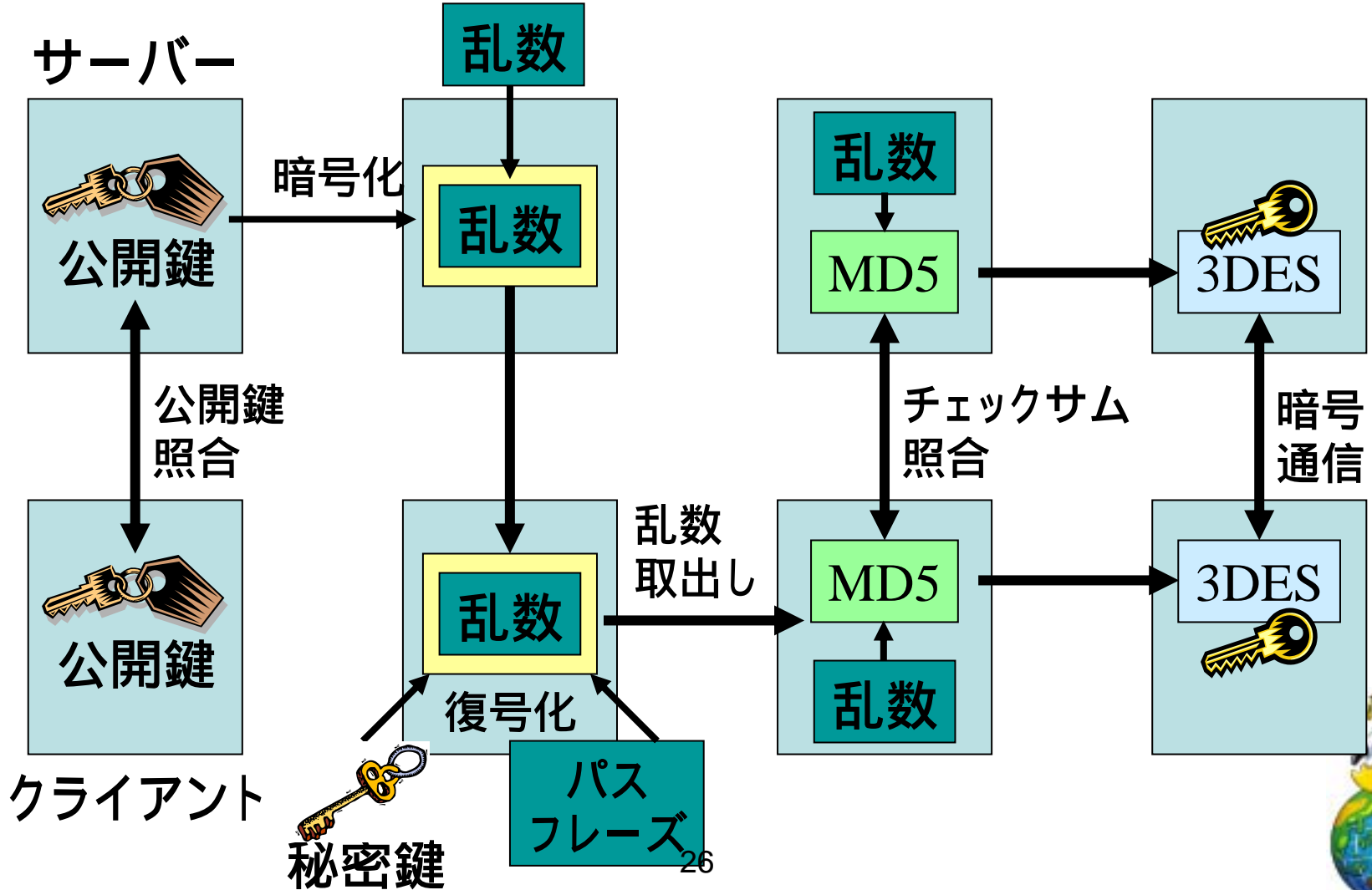


公開鍵・秘密鍵暗号

- 秘密鍵と公開鍵はペアで生成される
- 一方の鍵で暗号化されたデータは、もう一方の鍵だけが復号化できる



公開鍵認証



公開鍵・秘密鍵の生成

OpenSSHが使う公開鍵と秘密鍵のペアを生成

1. 鍵の生成にはssh-keygenコマンドを使用
 - -tオプションで使用する暗号化アルゴリズムを指定
 - ssh-keygen -t dsa (SSH v2・DSA暗号)
2. 実行時にはパスフレーズの入力が必要
 - 画面には表示されない
 - 2回入力が必要
3. 鍵は~/.sshディレクトリに生成される
 - 秘密鍵ファイル id_dsa
 - 公開鍵ファイル id_dsa.pub



公開鍵をサーバーに設置

1. 公開鍵を~/.ssh/authorized_keysに追加
 - 上書きを防ぐため、cpコマンドではなくcatコマンド + 追加リダイレクト(>>)で行うこと
 - 他人の公開鍵の場合には事前に「指紋」で鍵の同一性チェックを行う(ssh-keygen -l)
2. ~/.sshディレクトリとauthorized_keysの所有者とパーミッションの確認と変更
 - ~/.sshディレクトリ モード700(rwx-----)
 - authorized_keys モード600(rw-----)



まとめ

- じっくりとサーバ構築などのスキルを学ぶ
幹作り
- 関連するトピックに対する知識を増やす
枝葉を伸ばす
- 実機を使用して1つずつ確認しながら
 - manコマンドが重要
- 問題集などでスキルチェック



例題1

- カーネルのソースコードにパッチを当てるために使用するコマンドを2つあげなさい



例題2

PCMCIA (PCカード) 型の無線LANカードをシステムに追加しました。手動では利用できますが、システム起動時に自動的に設定されません。原因として適当と思われるものを選びなさい。

1. カードサービスが起動時に適切に設定されていない
2. 起動時スクリプトにPCMCIAの初期化が含まれていないか、適切な順序で初期化されていない
3. PCMCIAサポートはカーネル本体に組み込まれていない
4. カーネル起動時にHot-plugが有効になっていない



例題3

ext2ファイルシステムに不良ブロックがあるようです。ファイルシステムを破壊しない検査を行うコマンドをすべて選びなさい。

1. fsck
2. mke2fs
3. tune2fs
4. badblocks
5. e2fsck



例題4

ssh経由でユーザrootがログインする権限を設定しているファイルを選びなさい。

1. ssh.config
2. ssh_config
3. sshd.config
4. sshd_config



例題1の解説

- カーネルソースのパッチはgzip/bzip2で圧縮されています
 - 1ファイルで、tar形式ではありません
- 使用するコマンド候補
 - gzip/bzip2
 - gunzip/bunzip2
 - patch/patch-kernel



例題2の解説

- PCMCIA (PCカード)を使用するには pcmcia_csが必要
 - モジュールのロード
 - cardmgrの実行
- 1. 手動ならば動いているので設定は正しい
- 2. PCMCIAをネットワーク設定より先に初期化する必要がある (正解)
- 3. PCMCIAはモジュールとしてロード可能
- 4. Hot-plugはこの場合関係ない

http://www.lpi.org/en/tasks_201.html より出題



例題3の解説

- badblocksコマンドは不良ブロックを調べて、情報を書き出す (-oオプションで書き出し先のファイルを指定可能)
- fsckコマンドは不良ブロックを調べて、修復する (正解とされているが、デフォルト動作は修復を行うので、破壊的になるのではないか?)
- できるだけCDからレスキューモードで起動するなどして、デバイスを未使用状態で検査したい



例題4の解説

- OpenSSHはSSHクライアント(ssh)とSSHサーバ(sshd)の2つで構成されている
- 設定ファイルは/etc/sshディレクトリ内に
 - ssh_config クライアント設定ファイル
 - sshd_config サーバ設定ファイル(正解)
- rootのログインを許すかどうかはsshd_configに「PermitRootLogin」で設定

