JJS plus
New Style , Neo Alive.

Linux Professional Institute Japan LPI-JAPAN

LPICレベル2技術解説セミナー

2016/11/5

株式会社ジェイ・ジェイ・エス
研修教育事業部
大森　聡

講師プロフィール

JJS plus
New Style | Neo Alive.

会社概要

株式会社ジェイ・ジェイ・エス(http://www.jjs.co.jp/)

講師紹介

研修教育事業部所属　　大森　聡

Linuxおよびネットワーク・セキュリティ系をメインに、ベンダー系資格対策講習、情報処理技術者試験講座、新人SE教育研修など、様々な技術研修を担当

Linux Professional Institute

本日のアジェンダ

JJS+plus
New Style | Neo Alive.

- LPICレベル2　受験の心構え
  LPI認定資格とは
  試験の範囲
  学習環境の構築

- 技術的な詳細解説
  LPIC201試験範囲より頻出ポイントを解説

  休憩(10分程度)

  LPIC202試験範囲より頻出ポイントを解説

Linux Professional Institute

LPI認定資格とは

JJS plus
New Style Neo Alive.

- ベンダーニュートラルなグローバル認定資格
  NPOであるLinux Professional Instituteが実施

- Linux技術者のスキルを計る一つの目安
  他にRHCEなどのベンダー資格もある

- レベル1からレベル3までの3段階に分れる
  レベル1認定 - LPI 101,102試験の両方に合格
  レベル2認定 - レベル1認定に加え、LPI 201,202試験の両方に合格
  レベル3認定 – レベル2認定に加え、LPI 300, 303, 304試験のいずれかに合格

  ※詳細については、LPICのメインサイト(http://www.lpi.or.jp/)をご覧ください

Linux Professional Institute

201試験の概要

JJS+plus
New Style | Neo Alive.

- 出題数：60問
- 制限時間：90分
- 合格ライン：65%程度
- 試験範囲

  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

Linux Professional Institute

202試験の概要

JJS plus
New Style | Neo Alive.

- 出題数：60問
- 制限時間：90分
- 合格ライン：65%程度
- 試験範囲

  主題207：ドメインネームサーバ
  主題208：Web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

Linux Professional Institute

学習環境の構築

JJS plus
New Style | Neo Alive.

学習教材

市販のテキスト、問題集、Webサイトなどをベースに基礎知識を習得する

設定ファイル名、コマンド名、コマンドオプション、コマンドの実行結果、ログなどについては、実機でも確認する

実機環境

CentOS(Ver5),Feroda,Debian,Ubuntuなどのディストリビューションがおすすめ

メモリに余裕のあるマシン上に、Virtual Box, VM Player, Virtual PCなどの仮想化ツール(無償)を導入する

仮想化ツールを利用して、同時に複数台のLinuxを稼働させると、クライアント-サーバ系のアプリケーションの動作検証ができる

技術詳解の前置き

JJS plus
New Style | Neo Alive.

- LPICレベル2試験のポイントとなる部分について重点的に解説します

試験で出題される可能性の高いコマンド、設定ファイル・設定項目等

覚えておくべきLinuxの知識

学習する際に、混乱しやすい技術的な事柄

主題200：キャパシティプランニング

JJS plus
New Style Neo Alive.

- キャパシティプランニングとは？
  コンピュータシステムのリソース(CPU、メモリ、ディスク、ネットワーク帯域など)が将来的に不足しないようにあらかじめ設計すること。

- キャパシティプランニングにおける監視項目
  CPU使用率、メモリ使用率、ディスクI/O、スワップ使用率、ネットワークI/O

- 各リソース使用率を測定するためのコマンドおよびツール
  top, uptime, free, vmstat, lsof, sar/sadc, collectd

※学習ポイント：各コマンドでどのリソースの使用率を測定できるかまとめておく

- カーネルモジュールとは？
  カーネルを構成する機能を、モジュールとしてカーネル本体から分離したもの。ローダブルモジュールとも呼ぶ。おもにデバイスドライバがカーネルモジュールとして使用される。

- カーネルモジュールの管理コマンド
  lsmod, insmod, rmmod, modprobe, depmod, modinfo

- modprobeコマンド
  モジュールの依存関係を調べて、モジュールのロード/アンロードを実行する

- modules.depファイル
  モジュール間の依存関係が記述されたファイル。depmodコマンドで更新する

※学習ポイント：各コマンド、設定ファイルの役割、関連を正確に把握する

カーネルのコンパイルとインストール手順

1. カーネルソースを用意する(/usr/src/linux/)
2. カーネルコンフィグレーションを設定する(make configコマンド)
3. カーネル本体とカーネルモジュールをコンパイルする(makeコマンド)
4. カーネルモジュールをインストールする(make modules_install)
5. カーネル本体をインストールする(make install)
6. ブートローダ(GRUBまたはLILO)の設定を更新する

カーネルコンフィグレーションファイル

.configファイル　..　モジュール毎に、カーネルに静的に組み込む(Y)か、動的に組み込む(M)か、あるいは組み込まない(N)か指定する

make config, make menuconfig, make xconfig等のコマンドで編集が可能

※学習ポイント：カーネル構築の各段階でどのコマンドが必要なのか整理する

- Linuxの起動シーケンス(SysVinitの場合)
  1. BIOS(UEFI)の実行
  2. ブートローダ(GRUBまたはLILO)の実行
  3. カーネルのロード
  4. initプロセスの実行 → /etc/inittabファイルを参照
  5. 各種rcスクリプトの実行
  6. ランレベルに対応したサービスの実行

- サービスの自動起動(ランレベル3・SSHの場合)
  /etc/init.d/sshdのシンボリックリンクが/etc/rc3.d/S55sshdとして作成される。

- ブートローダの設定ファイルとインストールコマンド
  LILOの場合:/etc/lilo.conf, /sbin/lilo
  GRUB Legacyの場合:/etc/grub.conf(/boot/grub/menu.lst), grub-install

※学習ポイント:LPICレベル2では、Linux起動手順の詳細知識が問われる

/etc/fstabの役割

Linuxが利用するファイルシステムに関する情報が記述されており、システム起動時に参照される。

```
(例)/dev/sda1    /boot    ext3        defaults      1   2
    /dev/sr0     /media   iso9660     ro,noauto     0   0
        ①          ②        ③           ④         ⑤   ⑥
```

①デバイスファイル名　※ラベルやUUIDによる指定も可能
②マウントポイント
③ファイルシステムの種別
④マウントオプション
⑤dumpコマンドの対象
⑥fsckコマンドでチェックされる順序

※学習ポイント：/etc/fstabの書式および設定パラメータを正確におさえる

ファイルシステムの作成手順

1. パーティションの作成 → fdiskコマンド
2. ファイルシステムの作成 → mke2fsコマンド
3. マウント/アンマウント → mount / umountコマンド

ファイルシステムの保守

ファイルシステムのチェック → fsckコマンド
ファイルシステムのパラメータ変更 → tune2fsコマンド
ファイルシステムの詳細確認 → dumpe2fsコマンド

tune2fsコマンドのオプション

tune2fs -j → ext2からext3へ変換する
tune2fs -L → ファイルシステムのラベルを指定する
tune2fs -c → ファイルシステムのチェックを行うマウント回数を指定する

※学習ポイント：tune2fsマンドのオプションについては要注意

LVM(論理ボリューム管理)の特徴

既存のパーティションのサイズ変更やパーティションの移動、複数ディスクにまたがるパーティションの作成など、柔軟なパーティション管理が可能

LVMの作成手順

1. 従来のパーティションを物理ボリュームとして設定する(pvcreateコマンド)
2. 物理ボリュームからボリュームグループを構成する(vgcreateコマンド)
3. ボリュームグループの中から論理ボリュームを取得する(lvcreateコマンド)

作成した論理ボリュームは、従来のパーティションと同様に、mke2fsコマンドでファイルシステムを作成し、mountコマンドでマウントして利用することができる

※学習ポイント：LVM関連のコマンド群を整理しておく。LVM管理コマンドとして、vgextend / vgreduceコマンド(ボリュームグループの拡張/縮小)も要注意

主題205：ネットワーク構成①

JJS plus New Style Neo Alive.

- ネットワーク関連コマンド

ifconfig .. ネットワークインタフェースの表示および設定

arp .. ARPエントリの表示および登録・削除

tcpdump .. ネットワークインタフェースを通過するパケットのキャプチャ

nc .. TCP/UDPの通信状態の確認およびポートスキャンの実施

netstat .. 各種ネットワーク情報の表示

route .. ルーティングテーブルの表示および設定

※学習のポイント：ifconfig,arp,routeコマンドについては、ipコマンドで代用可能

- ネットワーク関連ファイル

/etc/hosts .. IPアドレスとホスト名の対応関係
(例) 192.168.0.1　　server.example.com　server

/etc/nsswitch.conf .. 名前解決を行う際の問い合わせの順序
(例) hosts:　files　dns

/etc/resolv.conf .. 問い合わせ先のDNSサーバとドメイン名
(例) search　example.com
　　nameserver 192.168.0.2

※学習のポイント：各種ネットワーク設定ファイルとその役割をおさえる

ソースアーカイブからのインストール手順

※software.tar.gzをインストールする場合

1. tar zxvf software.tar.gz　→解凍＆展開
2. ドキュメント(README,INSTALL等)の参照
3. ./configure　→Makefileの生成
4. make　→コンパイル
5. su　→root(管理者)権限を取得
6. make install　→インストール

※学習ポイント:インストール時に管理者権限が必要になる点に注意

休憩時間

JJS plus
New Style | Neo Alive.

10分ほど休憩いたします。

xx:xxに再開しますので、それまでにご着席ください。

Linux Professional Institute

DNSの実装

パッケージ名:BIND
デーモン名:named
設定ファイル:/etc/named.conf

named.confの設定例

```
acl localnet { 10.0.0.0/24; };  //アクセスリスト
options {
    directory "/var/named";
    allow-query { localhost; localnet; };  //アクセス制限
    allow-transfer { 10.0.0.2; };          //ゾーン転送の制限
 };

zone "." {  type hint;  file "named.ca";  };  //ルートヒントファイル
zone "example.com" { type master;  file "example.zone";  };       //正引きゾーンファイル
zone "0.0.10.in-addr.arpa { type master;  file "example.rev";  };   //逆引きゾーンファイル
```

※named.confについてはアクセス制限に関する設定が頻出

正引きゾーンファイルの例

```
$TTL 86400
@       IN      SOA     sv1.example.com.    root.example.com.   (
                        2016110101 ; Serial
                        10800      ; Refresh
                        600        ; Retry
                        3600000    ; Expire
                        86400 )    ; Negative TTL

        IN      NS      sv1.example.com.
        IN      NS      sv2.example.com.
        IN      MX 10   sv1.example.com.
        IN      MX 20   sv2.example.com.

sv1     IN      A       10.0.0.1
sv2     IN      A       10.0.0.1
host    IN      A       10.0.0.3
www     IN      CNAME   host.example.com.
```

逆引きゾーンファイルの例

```
$TTL 86400
@       IN      SOA     sv1.example.com.    root.example.com.   (
                        2016110101 ; Serial
                        10800      ; Refresh
                        600        ; Retry
                        3600000    ; Expire
                        86400 )    ; Negative TTL

        IN      NS      sv1.example.com.
        IN      NS      sv2.example.com.

1       IN      PTR     ns1.example.com.
2       IN      PTR     ns2.example.com.
3       IN      PTR     host.example.com.
```

※学習のポイント:ゾーンファイルの書式については詳細を理解しておく

Webサーバの実装

パッケージ名：Apache
デーモン名：httpd
設定ファイル：/etc/httpd/conf/httpd.conf

httpd.confの設定例(抜粋)

```
UserDir              public_html              #一般ユーザの公開ディレクトリ
ErrorLog             logs/error_log           #エラーログファイルの指定

AccessFileName       .htaccess                #外部設定ファイル
AllowOverride        AuthConfig Limit         #外部設定ファイルによる許可項目

SSLEngine            on                       #SSL機能の有効化
SSLCertificateKeyFile  /etc/pki/tls/private.key  #秘密鍵の指定
```

※学習ポイント：SSL関連のディレクティブ(設定項目)が頻出

プロシキサーバの実装

パッケージ名：Squid
デーモン名：squid
設定ファイル：/etc/squid/squid.conf
役割：コンテンツキャッシュ、アクセス制御

squid.confの設定例(抜粋)

```
http_port  8080                                    #squidが利用するポート番号

acl  localnet  src  192.168.0.0/255.255.255.0      #アクセスリストの定義
acl  blacklists  urlpath_regex  “/etc/squid/url_blacklist.txt”

http_access  allow  localnet                       #アクセスの許可/拒否
http_access  deny  blacklists
```

※学習ポイント：Squidのほかに**Nginx**に関する設定もみておく

ファイルサーバの実装①

パッケージ名:Samba
デーモン名:smbd(ファイル共有、認証), nmbd(ブラウジング、名前解決)
設定ファイル:/etc/samba/smb.conf

smb.confの設定例(抜粋)

```
unix password sync = yes | no          #WindowsとLinuxのパスワードを同期させるか否か
hosts allow = 10.0.0.0/255.255.255.0   #アクセス制御
browseable = yes | no                  #ブラウザリストに表示するか否か
```

※共有名の前に"$"をつけて、ブラウザリストで非表示にすることも可能

```
writable = yes | noまたはread only = no | yes      #ファイルの書き込みを可能にするか否か
security = share | user | server | domain | ad     #認証方式の指定
```

※学習ポイント:Windowsネットワークの基本的なしくみについておさえておく

ファイルサーバの実装②

パッケージ名:NFS
デーモン名:portmap, nfsd, mountd
設定ファイル:/etc/exports

NFSサーバ側(exportsファイルの設定例)

/share　10.0.0.0/255.255.255.0(rw, no_root_squash)　　#/shareディレクトリを10.0.0.0/24ネットワークに対して、読み書き可能で公開し、root権限でのアクセスを許可

※exportsファイルを記述した後、exportfsコマンドを実行して、設定を有効化する

NFSクライアント側(mountコマンドの実行例)

mount -t nfs server:/pub　/mnt/nfs　　#server上で公開された/pubを/mnt/nfsにマウントする

※学習ポイント:/etc/exportsファイルの書式が頻出事項

DHCPサーバの実装

パッケージ名：ISC DHCP
デーモン名：dhcpd
設定ファイル：/etc/dhcpd.conf

dhcpd.confの設定例(抜粋)

```
subnet 192.168.0.0 netmask 255.255.255.0 {
        option  routers                 192.168.0.1;            #デフォルトゲートウェイアドレス
        option  domain-name-servers     192.168.0.1, 192.168.0.2;       #DNSサーバアドレス
        range                           192.168.0.10  192.168.0.100   #アドレスプールの範囲
        host print-server {                                             #固定アドレス割り当て
                hardware ehternet aa:bb:cc:dd:ee:ff;
                fixed-address  192.168.0.200;
        }
}
```

※学習ポイント：dhcpd.confの設定項目について正確におさえておく

PAM(Pluggable Authentication Modules)

役割：Linuxの各種サービスに対し、一元的な認証機能を提供
設定ファイル：/etc/pam.d/内に、サービスごとに作成

設定ファイルの書式

モジュールタイプ	コントロール	モジュールのパス
auth	required	pam_nologin.so

※/etc/nologinファイルが存在した場合は、一般ユーザによるログインを拒否する設定

PAMの出題ポイント

コントロールの種類と特徴

requisite .. モジュールの実行に失敗したら、拒否
required .. モジュールの実行に失敗しても、同じタイプのモジュールをすべて実行した後で拒否
sufficient .. モジュールの実行に成功したら、許可

LDAPサーバの実装

パッケージ名:OpenLDAP
デーモン名:slapd
設定ファイル:/etc/openldap/slapd.conf(サーバ側)
設定ファイル:/etc/ldap.conf(クライアント側)
利用目的:アドレス帳、ユーザ認証データベースなど

主な属性

objectClass .. オブジェクトクラス　※エントリの特性を表す
dc .. domain component(ドメイン要素)
o .. organization(組織)
ou .. organizational unit(組織単位)
cn .. common name(一般名)
uid .. user id(ユーザID)

※上記の属性は、スキーマファイルによって定義され、それぞれにユニークな
OID(Object ID)が付与されている

メールシステムの概要

MTA(Mail Transfer Agent) .. SMTPによるメール転送処理を行う
MDA(Mail Delivery Agent) .. ローカルホストでメール配信を行う
MUA(Mail User Agent) .. メールクライアントソフト
POP/IMAPサーバ .. メールクライアントが接続してメールを受信する

MTAの実装

パッケージ名：Postfix
設定ファイル：/etc/postfix/main.cf → メール転送処理の設定
/etc/postfix/master.cf → 各プロセスの動作設定
/etc/aliases → エイリアス(メールアドレスの別名)の設定
※/etc/aliasesファイルを変更した後は、newaliasesコマンドで設定を反映させる必要がある

※学習ポイント：Postfixの設定ファイルに関する問題が頻出

Procmailについて

MDA実装の一つ。メールをローカル配信する際、一定のルール(レシピ)に基づいてフィリタリングする機能がある

設定ファイル：各ユーザのホームディレクトリ内の.procmailrc

.procmailrcの書式

:0 [フラグ] [:ロックファイル]
＊ 条件式
アクション

(設定例) メールのサイズが256000バイト以下の場合、/usr/bin/commandで処理する

```
:0
* < 256000
| /usr/bin/command
```

SSHの実装

パッケージ名:OpenSSH
デーモン名:sshd
設定ファイル:/etc/ssh/sshd_config

sshd_configの設定例(抜粋)

```
Protocol  2                    #sshのバージョン ※バージョン1には脆弱性あり
PermitRootLogin  yes | no#rootによる直接ログインの有効/無効
X11Forwarding  yes | no  #X11フォワーディング機能の有効/無効
DenyUsers | AllowUsers       #ユーザ単位のアクセス制限
```

公開鍵認証方式に関する設定

ssh-keygenコマンド　公開鍵・秘密鍵のペア鍵を生成
ユーザの公開鍵をsshサーバの~/.ssh/authorized_keysファイルに保存する

※学習ポイント:OpenSSH以外では、netfilter,vsftpd,fail2ban,OpenVPNなどが出題される可能性あり

質疑応答

以上で、本講座のコンテンツは終了になります。
ご静聴ありがとうございました。

この後、質疑応答の時間とさせていただきます。