

2章 Linuxのインストールとパッケージ管理

2-1. ハードディスクのレイアウト設計

2-1-1. パーティション分割

- ルートパーティションと異なるパーティションとして構成する領域の例
 - /home/ : 一般ユーザのホームディレクトリ
 - /var/ : メールボックス、Webディレクトリ、ログファイル
 - /boot/ : 起動用のファイル、カーネルイメージ
 - スワップ領域 : 仮想メモリ
- 通常のパーティションの場合、あとからパーティションのサイズを変更することはできないが、LVMパーティションを利用した場合は、後からサイズを変更できる。
 - LVM(logical volume manager)パーティション : 複数のハードディスクやパーティションにまたがった記憶領域をまとめて、単一の論理ボリュームとして扱う。
 - PV(物理V) -> VG(ボリュームグループ) -> LV(論理V)
- ルートパーティション内に構成すべきディレクトリ
 - /bin/ : 一般ユーザも実行できる基本コマンドを格納
 - /sbin/ : 管理者ユーザが実行できる管理者用のコマンドを格納
 - /etc/ : 設定ファイルを格納
 - /lib/ : ライブラリを格納
 - /dev/ : デバイスファイルを格納

2-2. ブートローダ

2-2-1. GRUB Legacyの構成

- GRUBには以下の2種類
 - GRUB Legacy (CentOS6 etc)
 - GRUB 2
- GRUB Legacy
 - 設定ファイルは「/boot/grub/menu.lst」
 - 起動処理に関する内容が記載されている
 - default : timeoutした場合に起動するエントリ
 - timeout : システムが自動起動するまでの秒数
 - splashimage : 起動時のGRUB設定画面の背景画像
 - hiddenmenu : 起動時、既定でGRUBメニューを非表示にする
 - title : 起動するエントリの名前
 - root : ルートデバイスの指定

- kernel : カーネルイメージの指定
- initrd : 初期RAMディスクの指定
- ルートデバイス : (hd0,1)
 - hd : ハードディスク
 - 1つめの数字 : ディスク番号(0が1台目)
 - 2つめの数字 : パーティション番号(0が1つめ)
- カーネルイメージファイル
 - /boot/下に「vmlinuz～」ファイルがあり、これをもとに起動する
- /proc/cmdline : 起動時のカーネルオプション部分が格納されている
- GRUB 2
 - 設定ファイルは「/boot/grub/grub.cfg」
 - 編集用ファイルは「/etc/default/grub」「/etc/grub.d」
 - 編集用ファイルを編集したあと、「grub-mkconfig」コマンドを実行
- GRUBはOSインストール時に起動ディスク領域に書き込むが、何らかの理由で欠損した場合は、以下で後から書き込むことができる

```
grub-install /dev/sda
```

2-3. 共有ライブラリの管理

2-3-1. 共有ライブラリ

これまで実行しているコマンドは、/bin/や/sbin/といったディレクトリに実行ファイルとして格納されているものが利用される。

これらの実行ファイルは、元となるプログラムソースがコンパイルされて生成されたもの。

ただこのファイルにすべての機能が含まれているわけではなく、**共有ライブラリ**を参照している。

```
ldd [ option ] ファイル #参照する共有ライブラリを表示する
```

2-3-2. 共有ライブラリへのパス指定

/lib/や/usr/lib/ディレクトリ以外に配置したい場合

- /etc/ld.so.confにディレクトリパスを記述し、「ldconfig」コマンドを実行する
- 環境変数LD_LIBRARY_PATHにディレクトリパスを追加する

※ldconfigコマンドを実行する理由は、プログラムが実際に参照するのが、/etc/ld.so.confではなく、/etc/ld.so.cacheのため。

```
ldconfig [ option(-p) ] [ ディレクトリ]
```

2-4. アプリケーション管理

2-4-1. ソースファイルからインストール

- ソースファイルからインストールする流れ
 - ./configure
 - configureスクリプト実行。Makefileを作成する
 - make
 - Makefileに基づいてコンパイルを実行。実行ファイルなどを生成。
 - make install
 - 生成されたファイルを所定のディレクトリに配置

2-4-2. パッケージファイルの利用

アプリケーション簡略化(依存関係などが複雑)のため、パッケージ管理システムがある。

- RPMパッケージ管理システム
 - RH、Fedora、CentOS
 - rpm
 - yum(CentOS 8からはdnf) ◎
 - /etc/yum.conf、/etc/yum.repos.d/ (リポジトリ情報)
- Debianパッケージシステム
 - Debian、Ubuntu
 - dpkg、dpkg-reconfigure
 - apt-get、apt-cache、aptitude ◎
 - /etc/apt/sources.list (リポジトリ情報)

2-4-3. Debianパッケージ管理システム

ローカルファイル/パッケージを利用したインストール

```
dpkg [option] [パッケージ名/ファイル名]
```

```
-i : インストール  
-r : アンインストール  
-p : 設定ファイルも含めてアンインストール  
-l : インストール済みパッケージの検索  
-L : インストールされたファイルの一覧表示  
-s : パッケージの情報を表示  
-S : ファイルのインストール元パッケージの検索
```

※RPM形式のパッケージがあるが、deb 1 形式がない。変換するエイリアンコマンド。

```
alien [option] [ファイル名]
```

```
-d : deb形式のパッケージファイルを生成  
-r : RPM形式のパッケージファイルを生成
```

```
dpkg-reconfigure [option] [パッケージ名]  
# パッケージのインストール時に設定を行うタイプのツールについて、このコマンドでやりなおす
```

リポジトリを利用したインストール

```
# debパッケージをリポジトリからインストールする  
apt-get サブコマンド [パッケージ名]  
  
update : パッケージ情報の更新  
install : パッケージのインストール  
upgrade : パッケージのアップグレード
```

```
# debパッケージをリポジトリから検索・参照する  
apt-cache サブコマンド [パッケージ名]  
  
search : パッケージの検索  
show : パッケージ情報の表示
```

```
# debパッケージをリポジトリから操作する（↑のをまとめて！）  
aptitude サブコマンド [パッケージ名]  
  
update : パッケージ情報の更新  
install : パッケージのインストール  
upgrade : パッケージのアップグレード  
search : パッケージの検索  
show : パッケージ情報の表示
```

2-4-4 RPMパッケージ管理システム

ローカルファイル/パッケージを利用したインストール

```
rpm [option] [パッケージ名/ファイル名]  
  
-i : インストール  
-U : アップグレード/インストールされていない場合はインストール  
-F : アップグレード/インストールされていない場合でもインストールしない  
-e : アンインストール  
-q : パッケージの検索  
  
#併用する  
-v : 詳細情報の表示  
-h : '#'で進行情報を表示
```

```
--nodeps : 依存関係を無視する
```

```
#-qとの併用
```

```
-a : all
```

```
-i : パッケージ情報の表示
```

```
-l : ファイルの一覧表示
```

```
-f : ファイルのインストール元パッケージの検索
```

```
-p : パッケージファイルを指定して検索
```

```
--changelog : 更新履歴を表示
```

```
--configfiles : パッケージに含まれる設定ファイルを一覧表示
```

リポジトリを利用したインストール

※apt-getコマンドと同様のyumを利用！！

```
#稼働しているシステムに関する情報を出力する
```

```
uname [-a]
```

```
#リポジトリからRPM形式パッケージのダウンロードを行う
```

```
yumdownloader パッケージ名
```

```
#インストールをするなら
```

```
yumdownloader localinstall パッケージ名
```