

2007年2月9日(金)
LPIC スタートアップ無料セミナー@東京

“オープンソース教育100年構想”
リナックスアカデミー

ファイルシステム管理とLVM

リナックスアカデミー 学校長

濱野 賢一郎

hamano@linuxacademy.ne.jp



本ドキュメントは、Linuxを中心とするオープンソースコミュニティの成果をもとに作成されています。開発者をはじめとするコミュニティの皆様には感謝いたします。

Linuxは、Linus Tovalds氏の米国およびその他の国における登録商標または商標です。

UNIX、X Window Systemは、The Open Groupの登録商標です。

Red Hat、RPMの名称は、Red Hat,Incの商標です。

Intel、Pentiumは、Intel Corporationの登録商標または商標です。

Microsoft、MS-DOS、Windowsは、米国Microsoft Corporationの米国およびその他の国における登録商標です。

LPIのロゴおよびLinux Professional Instituteという名称は、The Linux Professional Institute,Incの登録商標です。

その他記載された会社名およびロゴ、製品名などは該当する会社の商標または登録商標です。また、本文中では、(c)、(R)、TMの表示を省略しています。ご了承ください。

(C) 2007 Linux Academy. All rights reserved.

講師紹介

濱野 賢一郎（はまの けんいちろう）

- リナックスアカデミー 学校長
- 日本OSS推進フォーラム 人材育成部会
- 独立行政法人 情報処理推進機構（IPA）OSSセンター 人材育成WG 主査
- 経済産業省 ITスキルスタンダードLinuxプロジェクト 実施担当
- LPI認定試験研究会 発起人（の一人）
- 日本Apacheユーザ会／日本Sambaユーザ会 スタッフ
- 日本PHPユーザ会 発起人（の一人）
- 「Oracle for Linux メーリングリスト」「WebDAV-JP ML」管理人
- 著書に「Linux教科書 LPIC」「合格Expert LPI Linux認定試験」「PHP プログラミング Black Book」「qmailで作る快適メールサーバー」、翻訳書に「PHPデスクトップリファレンス」などがある。



最近では、Linuxやオープンソースソフトウェアを理解した技術者を養成するための体系的な仕組みを模索中。

■リナックスアカデミー

リナックスアカデミーは、オープンソースの専門教育機関として2001年に設立されたスクール。Linux、ネットワーク、Java、PHP、データベースを中心とするIT技術習得によるスキルアップ、就転職や業務転換などのキャリアアップ支援の総合サービスを提供している。

2005年度までに約3000名の方をIT業界に輩出しており、LPIC-1の合格者を1250名以上、LPIC-2の合格者を300名以上輩出するなど、圧倒的な資格合格実績をほこっている。

リナックスアカデミー・スクール
LAオープンソースユニバーシティ

<http://www.linuxacademy.ne.jp/>
<http://osu.linuxacademy.ne.jp/>

■日本OSS推進フォーラム

システムベンダーやユーザー企業などにより構成される非営利組織。オープンソースソフトウェア(OSS)の普及・促進に取り組むべく、OSSに関する様々な課題について議論し、それらを解決するための取り組みを行っている。2006年5月現在、「サーバー部会」「デスクトップ部会」「人材育成部会」という3つの部会があり、各部会内に設置されたタスクフォースで活動が行われている。 [出典:OSS iPedia]

■LPI認定試験研究会

LPI認定試験研究会(LPIC研究会／LPIC Study Group)は、

- LPI認定試験情報を提供し、LPI認定の普及促進を図る
- LPI認定試験対策を通してLinuxスキルの向上につとめる
- LPI認定試験関連のセミナー開催や記事執筆を行う
- Linuxユーザー同士の交流の場を作る

といった活動を行っています。活動は主にメーリングリストを通じて行われますが、関東や関西でのオフラインミーティングも定期的に行っています。

LPI認定試験研究会のWebサイト <http://www.lpi.jp/>

ディスクを利用するまでの手順

■ パーティションの作成

- ディスクを複数の領域に分割して管理できる
- 分割した領域単位をパーティションという
- fdiskコマンドなどを利用

■ ファイルシステムの作成

- パーティションに作成することによって、実際にファイルの配置が可能となる
- mkfsコマンドなどを利用

■ ファイルシステムのマウント

- ディレクトリツリーにファイルシステムをマウントすることが利用できる
- mountコマンドなどを利用



■ 手順例

I. パーティションの作成

1. fdiskコマンドを起動する
fdisk /dev/hda
2. 現在のパーティションテーブルを確認する
p
3. パーティションを作成する(50MBの論理パーティションの場合)
n → <Enter> → +50M
4. 再度現在のパーティションテーブルを確認する。新しいパーティション確認できる。
5. パーティションテーブルに書き込んでfdiskを終了する
w
6. (必要に応じて)システムを再起動する
shutdown -h now

II. ファイルシステムの作成

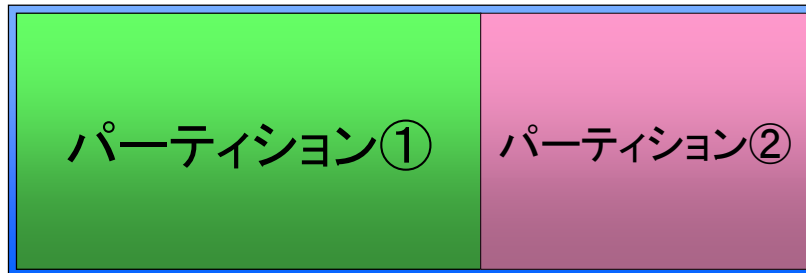
7. パーティション/dev/sd5にファイルシステムext3を作成する
mke2fs -j /dev/hda5

III. ファイルシステムのマウント

8. マウントポイントとして/testを作成する
mkdir /test
9. /dev/sda5を/testとしてマウントする
mount -t ext3 /dev/hda5 /test
10. dfコマンドを用いて確認する
df -h

およそのイメージ

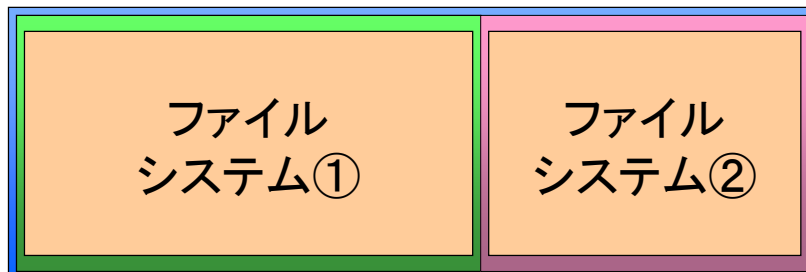
1. パーティションの 作成



およそのイメージ

1. パーティションの 作成

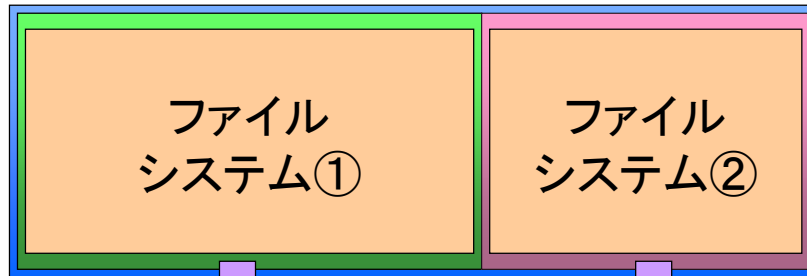
2. ファイルシステム の作成



およそのイメージ

1. パーティションの
作成

2. ファイルシステム
の作成



3. マウント

/home

/var



パーティションの作成

■ パーティション

- ディスク領域を分割する論理的な概念
- 基本パーティション、拡張パーティション、論理パーティションに分類できる
- 実際にファイルシステムが作成できるのは、基本パーティションと論理パーティション

■ パーティションの作成

- fdiskコマンドを用いることがパーティションの確認、作成が行える



■パーティション

1台のディスクドライブを複数の論理的な区画(パーティション)に分割して使うことができます。それぞれのパーティションには、異なるファイルシステムを作成できます。パーティションの種類は3つあります。

パーティションの実態は、ハードディスク上の領域の開始位置と終了位置の情報です。ディスクの先頭領域(のパーティションテーブル)に、各パーティションに対応する開始位置・終了位置のアドレスが格納されている。

■基本パーティション

ディスクには最大4個の基本パーティションを作成することができます。パーティション内にはファイルシステムが格納されます。基本パーティションのデバイスファイル名は、プライマリ・マスターに接続されたIDEハードディスクの場合、/dev/hda1～hda4となります。

■拡張パーティション

基本パーティションの1つを拡張パーティションにすることができます。拡張パーティションの中にはファイルシステムではなく、論理パーティションが格納されます。

■論理パーティション

論理パーティションとは、拡張パーティション内に作成されたパーティションのことです。論理パーティションは最大59個まで作成できます。デバイスファイル名は、プライマリ・マスターに接続されたIDEドライブの場合、/dev/hda5～hda63となります。

■fdisk コマンド

パーティションの作成、削除、変更、情報表示などを行います。-lオプションを指定すると、そのデバイス(ハードディスク)のパーティションテーブルの状態を表示します。

ファイルシステムの作成

■ ファイルシステムタイプ

- Linuxでは、ext3 の利用が標準的である
- ext3 ではジャーナリング機能が実現されている

■ ファイルシステムの作成

- mkfsコマンドで作成できる



■ ファイルシステムとファイルシステムタイプ

fdisk などでパーティションを作成しただけでは、そのパーティションを利用することはできません。Linuxでそのパーティションを利用するためには、そのパーティション上にファイルシステムを作成します。ファイルシステムとは、OSがファイルを管理するための枠組みであり、Linuxではext2、ext3、XFS、ReiserFSなどのファイルシステムが利用できます。

新しいパーティション上にファイルシステムを作成すると、メタデータと呼ばれる管理情報がパーティション内に作成され、そのパーティションを利用することが可能となります。

ext3、XFS、ReiserFS などのファイルシステムはジャーナリングファイルシステムと呼ばれ、ジャーナリング機能を持っています。ジャーナリング機能は、ファイルシステムの信頼性を向上させるための機能の1つです。ジャーナリングファイルシステムにおける「ジャーナル (Journal)」とは、ファイルシステムの変更に対する操作を、あらかじめ準備された領域にログとして記録することを意味します。ジャーナリングファイルシステムは、障害からの復旧時にジャーナルの情報を利用してファイルシステムの復旧を行い、ファイルシステムの一貫性を保つことができます。

■ ファイルシステムの作成

mkfsコマンドなどを用いると、パーティション上にファイルシステムを作成できます。mkfsコマンドは、各ファイルシステムの種類に対応したプログラムのフロントエンドであり、ファイルシステムの種類に対応したコマンドを呼び出します。たとえば、ext2ファイルシステムを使っている場合は、mkfs.ext2プログラムが呼び出されます。

ファイルシステムのマウント

■ マウント

- ファイルシステム内のディレクトリツリーを、特定のディレクトリ以下に結合すること
- マウントしたファイルシステムが結合されるディレクトリをマウントポイントという

■ マウント操作

- mountコマンドでマウントを行える
- umountコマンドでマウントを解除できる



■ ファイル/etc/fstab

ファイルシステムの情報は、/etc/fstabファイルに記述されています。マウントするときにはこの設定ファイルが参照されるため、マウントする頻度の高いファイルシステムを記述しておきます。

/etc/fstabの設定は、1つのファイルシステムごとに1行で設定を行い、各行設定は順に

- デバイスファイル名
 - マウントポイント
 - ファイルシステムタイプ
 - マウントオプション
 - dumpフラグ
 - ブート時にfsckがチェックする順序
- となっています。

```
/dev/hda5 /project ext3 defaults 1 2
```

従来のパーティションの限界

- 大容量のパーティションが作成できない
 - ・ ディスクサイズを超えることができない
- 原則として、パーティションサイズを変更できない
 - ・ 容量が不足した場合の対応が複雑になる

など難点がいくつか存在する・・・

これらの問題を解決するのが

LVM

(Logical Volume Manager)



■パーティションサイズの限界

パーティションはディスク領域を分割するものであるため、元のハードディスク容量以上のサイズのパーティションを作成することはできません。最近では、500GBや750GBなどのディスクが登場し、近いうちに1TBのディスクが流通するといわれていますが、それでも2TBや3TBなどのパーティションを作成することはできません。

しかし、画像や動画データなどをはじめとして大きなデータサイズを扱う業種におけるファイルサーバでは、数TB～数10TBなどの容量が当然のものとなっています。

■ファイルシステム容量不足時の対応

サーバを容量していると、多くの場合ディスクの空き容量が確実に少なくなります。データの増加やログの蓄積などによることが多いといえます。

ファイルシステムの空き容量が少なくなってきた時は、原則としてパーティションのサイズは変更できないため、十分なサイズを持つ新しいディスクもしくはパーティションを確保し、それにデータを移動させた上で、新しいパーティションをマウントするといった作業を行います。この作業は複雑である上に時間のかかる作業です。

また、ディスクから異音がするといった場合に、ディスクを交換するようなケースでも同様の作業を必要とします。

LVM (Logical Volume Manager)

■ LVM

- ディスク／パーティションを仮想化する仕組み
- ユーザーやソフトウェアから物理的なディスクを意識することなく利用できる
- ディスク容量不足時の対応(ディスクの増設)やディスクの交換などが容易に行える



■ LVM

LVM(Logical Volume Manager)は、ユーザーが扱うパーティションとして、論理ボリュームと呼ばれる単位でパーティションを提供して、物理的なディスクの存在を仮想化(Virtualization)します。その結果、物理的なディスクの増設や変更などが、ユーザーやアプリケーションに対して隠蔽されて、ディスクデバイス管理の柔軟性を向上させます。

LVMは、物理ボリューム(PV:Physical Volume)、ボリュームグループ(VG:Volume Group)、論理ボリューム(LV:Logical Volume)から構成されます。

LVMの構築

■ パーティションタイプ

- LVMでは「8e」を指定

■ 物理ボリュームの作成

- /dev/hda6~/dev/hda8に作成

■ ボリュームグループの作成

- vg01とする
- PVは/dev/hda6と/dev/hda7の2つとする

■ 論理ボリュームの作成

- lv01とする
- vg01から80MBを切り出す



■ 手順例

1. fdiskコマンドを用いて、/dev/hda6~/dev/hda8のパーティションタイプを「8e」(LVM用)に変更する。
t → (パーティション選択) → 8e
w
2. (必要に応じて)システムを再起動する。
shutdown -r now
3. (必要に応じて) カーネルモジュール dm-mod をロードする。
modprobe dm-mod
4. 物理ボリュームを作成する。
vgscan
pvcreate /dev/hda6
pvcreate /dev/hda7
pvcreate /dev/hda8
5. 物理ボリュームを確認する。
pvscan
6. ボリュームグループ(vg01)を作成する
vgcreate vg01 /dev/hda6 /dev/hda7
7. ボリュームグループを確認する。
vgscan
8. 論理ボリューム(lv01)を作成する。デバイスファイル/dev/vg01/lv01が作成される。
lvcreate -L 80M -n lv01 vg01

LVのサイズ変更

■ LVのサイズ変更

- lvextendコマンドでサイズを拡張できる
- VGの空き領域を用いる

■ ファイルシステムレベルでの調整

- ext3では resize2fsコマンドでスーパーブロックなどを再調整できる



■ 手順例

1. /dev/vg01/lv01をアンマウントする。
umount /mnt
2. 論理ボリュームlv01のサイズを変更する。
lvextend -L +15M /dev/vg01/lv01
3. fsckコマンドを用いてファイルシステムの整合性を確認する。
fsck -f /dev/vg01/lv01
4. resize2fsコマンドを用いてファイルシステムの情報を更新する。
resize2fs -p /dev/vg01/lv01
5. /mntをマウントポイントとしてマウントする。
mount -t ext3 /dev/vg01/lv01 /mnt
6. dfコマンドを用いてマウントされている事を確認し、サイズも確認する。
df -h

■ VGにPVを追加する

VGに空きPEがない場合には、新たにPVをVGに加えることでVGに空きPEを確保する。

ボリュームグループvg01に物理ボリュームPVを追加する。

```
# vgextend vg01 /dev/sda8
```

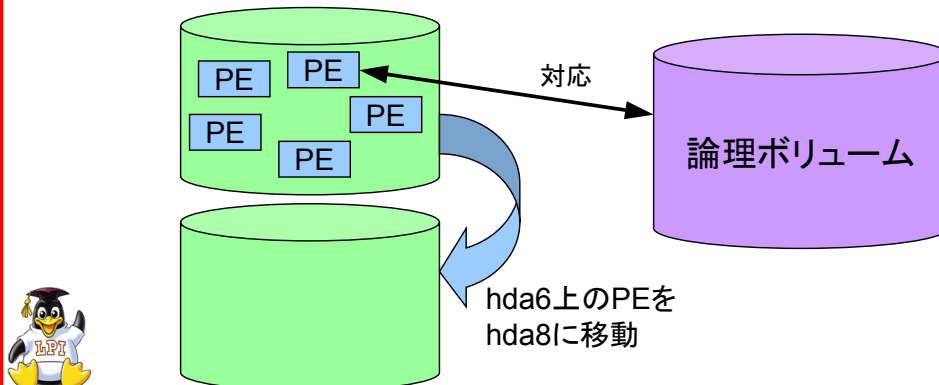
ボリュームグループの情報を参照する。

```
# vdisplay vg01
```

[参考] LVMにおけるディスク交換

■ /dev/sda6を解放する

- vg01に所属している/dev/hda6にあるエクステントを
あいている/dev/hda8に移動



■手順例

1. /dev/hda6、/dev/hda8のエクステントの状況を確認する。
pvdisplay /dev/hda6
pvdisplay /dev/hda8
2. /dev/hda6にあるエクステントを/dev/hda8に移動する。
pvmove /dev/hda6 /dev/hda8
3. 再度 1. を実行し、それぞれのエクステントの状況を確認する。
4. /dev/hda6をvg01から解放する。
vgreduce vg01 /dev/hda6
5. pvscanを用いてPVの状況を確認する。
pvscan
6. /dev/hda6をPVから解放する。
pvremove /dev/hda6
pvscan

学習のための参考情報

■ 書籍

- 『Linux教科書 LPIC レベル1 第2版』
中島 能和・濱野 賢一朗 著／翔泳社
- 『合格Expert LPI Linux 認定試験』
濱野 賢一朗・中島 能和 著／技術評論社

■ 研修

- リナックスアカデミー
 - 業界未経験者向けの総合教育サービス
 - <http://www.linuxacademy.ne.jp/>
- LAオープンソースユニバーシティ
 - ワンテーマ型の短期研修サービス
 - <http://osu.linuxacademy.ne.jp/>

