

LinuC レベル1 Version10.0 技術解説無料セミナー

2020/12/12 開催

主題	1.05 ハードウェア、ディスク、パーティション、ファイルシステム
副題	1.05.2 ハードディスクのレイアウトとパーティション

本日の講師

三澤 康巨

三澤 康巨

- KDDI株式会社で、電話等のネットワークサービス設備のエンジニアリングなど様々な業務を担当しました。
- 2017年10月から2年半、KDDIグループ内のサーバ研修講師を務めました。
- サーバ研修受講者の中から、200名を超える LinuC レベル1 合格者を出しました。
- 2020年3月、KDDIを定年退職しました。
- 2020年7月18日、LinuCレベル1 Version10.0 技術解説セミナー「ブートプロセスとsystemd」の講師を担当しました。
- 2020年11月28日、オープンソースカンファレンス2020 オンライン/福岡「Linuxマシンを作ってみよう ～LinuC レベル1 /レベル2 学習環境構築ガイド～」の講師を担当しました。

■LinuCとは

クラウド時代の即戦力エンジニアであることを証明するLinux技術者認定資格

- ✓現場で「今」求められている新しい技術要素に対応
 - オンプレミス／仮想化を問わず様々な環境下でのサーバー構築
 - 他社とのコラボレーションの前提となるオープンソースへの理解
 - システムの多様化に対応できるアーキテクチャへの知見
- ✓全面的に見直した、今、身につけておくべき技術範囲を網羅

今となっては使わない技術やコマンドの削除、アップデート、新領域の取り込み
- ✓Linuxの範疇だけにとどまらない領域までカバー

セキュリティや監視など、ITエンジニアであれば必須の領域もカバー

■Version10.0と従来の出題範囲の比較

	テーマ	Version 10.0	改訂前
LinuC レベル1	仮想化技術	<ul style="list-style-type: none"> ・仮想マシン／コンテナの概念 ・クラウドセキュリティの基礎 	← (Version10.0で新設)
	オープンソースの文化	<ul style="list-style-type: none"> ・オープンソースの定義や特徴 ・コミュニティやエコシステムへの貢献 	← (Version10.0で新設)
	その他	→ (Version10.0で削除)	アクセシビリティ、ディスククォータ、プリンタの管理、SQLデータ管理、他
LinuC レベル2	仮想化技術	<ul style="list-style-type: none"> ・仮想マシンの実行と管理(KVM) ・コンテナの仕組みとDockerの導入 	← (Version10.0で新設)
	システムアーキテクチャ	<ul style="list-style-type: none"> ・クラウドサービス上のシステム構成 ・高可用システム、スケーラビリティ、他 	← (Version10.0で新設)
	その他	<ul style="list-style-type: none"> ・統合監視ツール(zabbix) ・自動化ツール(Ansible) 	← (Version10.0で出題範囲に追加)
		→ (Version10.0で削除)	RAID、記憶装置へのアクセス方法、FTPサーバーの保護、他

ハードディスクのレイアウトとパーティション

- 1.はじめに
- 2.パーティションの概要
- 3.パーティション分割とディレクトリ構成
- 4.デバイスファイル
- 5.パーティションの作成
- 6.LVMの概要

1.はじめに

2.パーティションの概要

3.パーティション分割とディレクトリ構成

4.デバイスファイル

5.パーティションの作成

6.LVMの概要

- 本セミナーでは、LinuxシステムでHDDやSSD（以降、「ディスク」と総称）の初期設定や管理をする際に必要となる、パーティションに関する知識と、パーティションを作成・操作する方法について学習します。
- 学習効果を高めるため、実行例の出てくる部分では、ご自分でも実行してみることをお勧めします。
- Linuxには多数のディストリビューションが存在しますが、本セミナーの実行例では、CentOS 7 を使用します。
 - ビジネス用サーバーの多くで稼働している Red Hat Enterprise Linux 7 (RHEL7) と互換性があります。
 - RHEL7は有料ですが、CentOS 7 は無料で利用できます。
 - CentOSとRHELの最新バージョンは「8」ですが、本セミナーでは、普及が進んでいるバージョン「7」を使います。

- CentOS 7 に基づく学習環境の構築方法を、LPI-Japanのサイトでご紹介しています。

LinuC レベル1 / レベル2 Version 10.0 学習環境構築ガイド

https://linuc.org/docs/v10/guide_text.pdf

- 学習環境構築ガイドでは、2種類の環境の構築方法を紹介しています。

【環境A】

- 用意したコンピュータの内蔵ストレージを上書きして、Linux専用コンピュータを構築します。
- WindowsやMacOS等の既存OSは使えなくなります。
- 不要になった古いPC等がある場合に、それを使ってください。

【環境B】

- WindowsやMacOS等の既存OSを壊すことなく、外付けSSDにLinuxをインストールします。
- これによって、既存OSとLinuxとの間を切り替えて利用することができます。
- 但し、既存OSとLinuxとを同時に利用することはできません。

- 本セミナーの実行例を実習するためには、学習環境構築時にGPTパーティションを手動で設定し、ディスクに空き領域を残しておくことが必要です。

1.はじめに

2.パーティションの概要

3.パーティション分割とディレクトリ構成

4.デバイスファイル

5.パーティションの作成

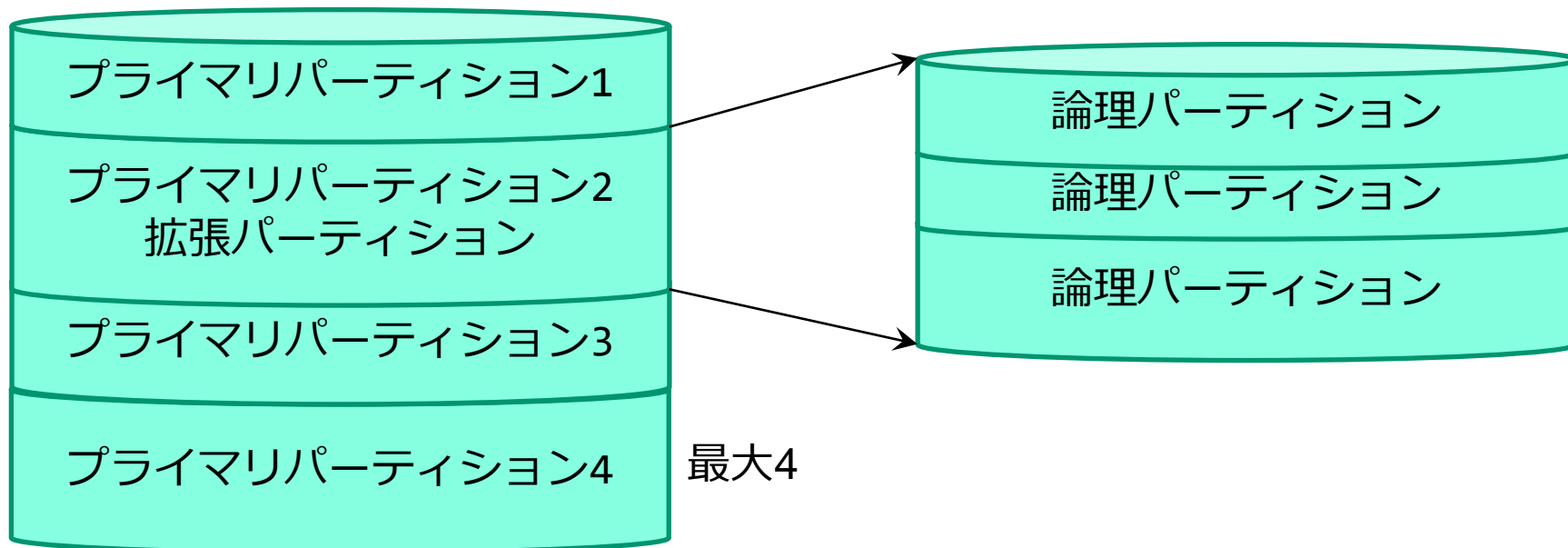
6.LVMの概要

- Linuxではディスク装置を分割して利用します。分割された各部分を「パーティション」と呼びます。
- システムにディスク装置を接続して、利用できるようにするには、パーティションに分割し、パーティション毎にファイルシステムを作成し、ルートファイルシステムにマウントする必要があります。

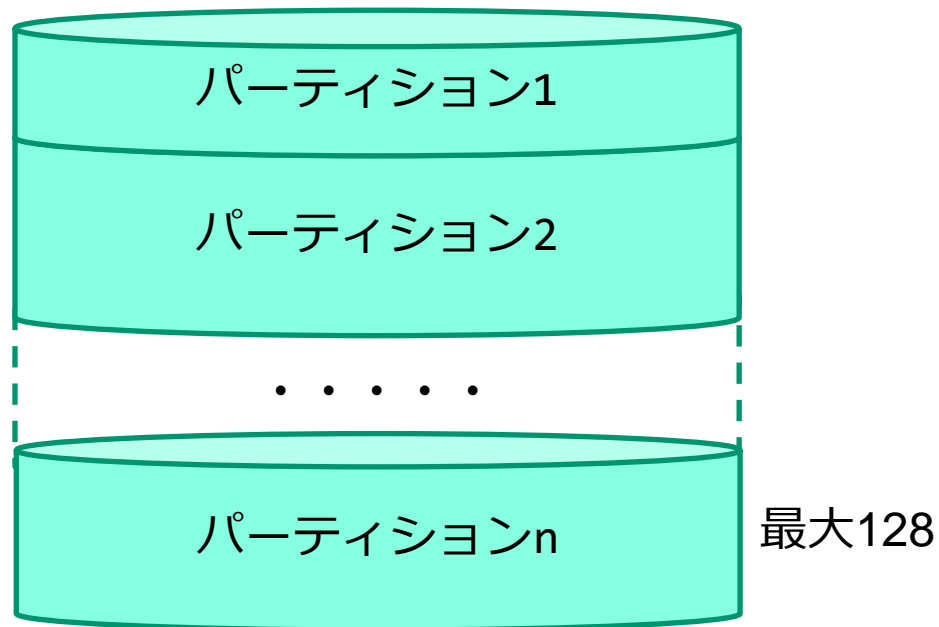


- 本日は、パーティション（副題 1.05.2）について学習します。

- パーティションの規格には、古くからあるMBR (Master Boot Record) と新しいGPT (GUID Partition Table) とがあります。
- MBRパーティションの場合、1つのディスクに最大4つのパーティションを作成でき、これをプライマリパーティションと呼びます。5つ以上に分割したい場合は、プライマリパーティションの1つを拡張パーティションに指定し（1つだけ）、その中を論理パーティションに分割します。



- MBR方式で利用できるディスクは最大2TiBです。それを超える大容量ディスクを利用するには、GPT方式を使う必要があります。
- GPT方式の場合には、1つのディスクに最大128個のパーティションを作成することができます。プライマリパーティション、拡張パーティション、論理パーティションの区別はありません。



- 本セミナーでは、主にGPT方式で説明します。実行例もGPTパーティションを扱います。

- Linuxシステムの電源をONすると、以下のプロセスでシステムが起動します。
(副題 1.01.3の出題範囲)



- 電源ONすると、BIOSまたはUEFIが起動します。BIOS (Basic Input/Output System) はコンピュータ内のROMに書き込まれた最も原始的なプログラム (ファームウェア) です。BIOSはブートデバイス (HDD、SSD等) を選択して、そこに格納されているブートローダを起動します。UEFI (Unified Extensible Firmware Interface) はBIOSを改良したファームウェアで、システムを起動するための機能はBIOSと同じです。
- ブートローダはカーネルを選択して起動するプログラムで、GRUB LegacyとGRUB2の2種類があります。
(ここまで、副題 1.01.3の出題範囲)
- BIOSはMBRとGPTの両パーティション規格に対応していますが、UEFIはGPTのみを扱います。
- MBRパーティションの場合、ブートローダは起動デバイスの先頭にあるMBR (Master Boot Record) 領域に格納されています。
- GPTパーティションの場合には、ブートローダは起動デバイスのESP (EFI System Partition) に格納されています。ESPは /boot/efi にマウントされます。

1.はじめに

2.パーティションの概要

3.パーティション分割とディレクトリ構成

4.デバイスファイル

5.パーティションの作成

6.LVMの概要

- Linuxのファイルシステムでは、一般的にルートディレクトリの下に多数のディレクトリがあります。それらを一つの「ルートパーティション」として構成することも可能ですが、以下の理由から、複数のパーティションに分割することが推奨されています。
 - 障害発生時に被害が波及する範囲を限定することができる。
 - 特定のユーザが過大にディスク容量を使用した場合でも、その影響を当該パーティション内に限定し、システム全体への影響を防ぐことができる。
 - パーティション毎にバックアップを取ることができる。
- ファイルシステム用パーティション以外に、スワップ領域として使われる「スワップパーティション」が必要です。メモリが不足すると、メモリ内容の一部をスワップ領域に退避することによって、システムダウンを防いで、処理を継続します。

- Linuxの動作仕様上、ファイルシステム内のディレクトリ名称に制約はありません。しかし、マシン毎に異なるディレクトリ名が使われていると、ユーザーには不便です。そこで、ディレクトリの用途と名称との間にガイドラインが設けられていて、FHS (Filesystem Hierarchy Standard) と呼ばれます。
- ルートパーティションとスワップパーティションの2つがあれば、システムは動作することができます。しかし、前記の理由から、少なくとも /boot、/home、/var、/tmp を個別のパーティションに分割することが一般的です。

ディレクトリ	用途
/boot	カーネルイメージ等のシステム起動に必要なファイル
/bin	一般ユーザー用のコマンド
/sbin	システム管理用のコマンド
/home	ユーザーのホームディレクトリ
/root	rootユーザー（管理者）用ホームディレクトリ
/var	ログやメールなどの可変データ
/tmp	一時ファイル

続く

続き

ディレクトリ	用途
/lib	ライブラリプログラム
/opt	追加アプリケーション
/etc	各種設定ファイル
/usr	ユーザーコマンド、ソースコード等
/dev	デバイスファイル
/proc	システム状態を参照する仮想ファイル
/media	リムーバルメディア用のマウントポイント
/mnt	一時マウントポイント

- 1.はじめに
- 2.パーティションの概要
- 3.パーティション分割とディレクトリ構成
- 4.デバイスファイル**
- 5.パーティションの作成
- 6.LVMの概要

- Linuxでハードウェアを指定するには、それぞれに対応したファイル名を使います。これらをデバイスファイルと呼び、 /dev ディレクトリの下に置かれています。
- 例えば、CD/DVDドライブのデバイスファイルは /dev/sr0 （2台目は sr1）です。
- ディスクのデバイスファイル名は接続されているインタフェースによって異なりますが、大半で使われているSCSI系のディスクでは、以下のとおりです。
 - /dev/sda （1台目）
 - /dev/sdb （2台目）
 -

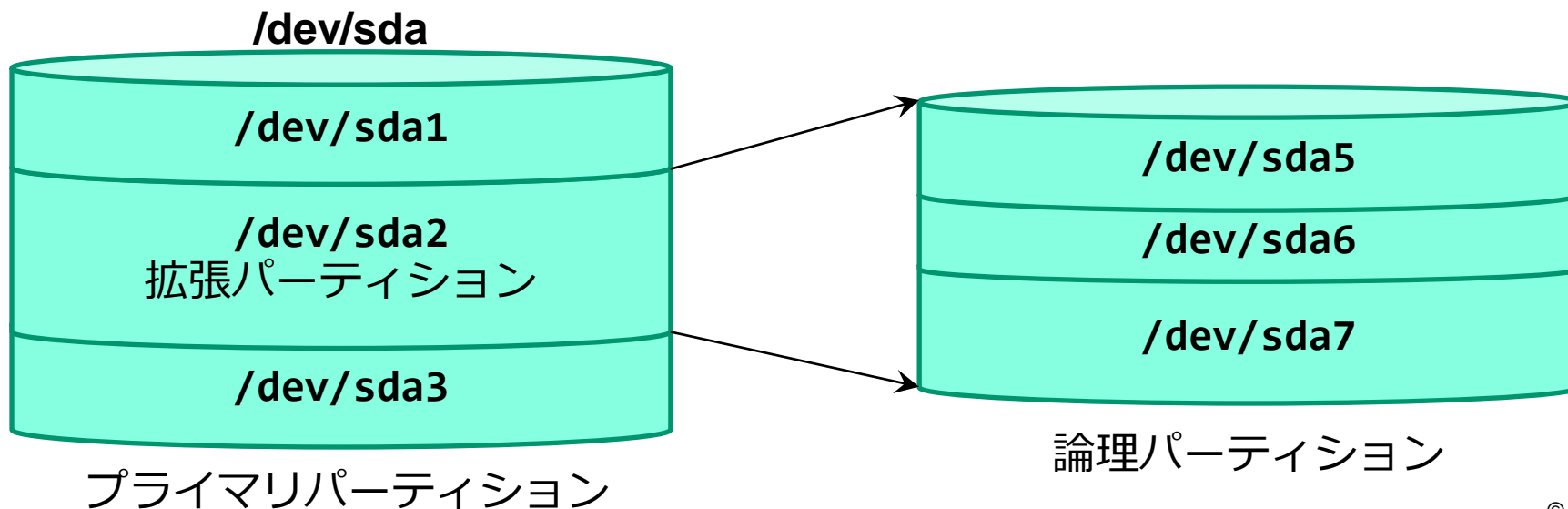
- パーティションのデバイスファイル名は、ディスクのデバイスファイル名の後にパーティション番号を追加した名称です。例えば、1台目のSCSI系ディスク上のパーティション名は、以下のとおりです。

/dev/sda1 (1つめ目のパーティション)

/dev/sda2 (2つめ目のパーティション)

.....

- MBRパーティションの場合は、プライマリパーティションが sda1~sda4、論理パーティションが sda5~ (必ず5から) です。



- 1.はじめに
- 2.パーティションの概要
- 3.パーティション分割とディレクトリ構成
- 4.デバイスファイル
- 5.パーティションの作成**
- 6.LVMの概要

- パーティションの作成、設定変更、削除には、以下のコマンドを使用します。どれを使っても、同様の操作が可能です。

parted

gdisk

fdisk

- partedコマンドとgdiskコマンドは、MBRとGPTの両方に使用できます。
- fdiskコマンドは、MBR専用のコマンドです。（但し、「実験フェーズ」としてGPT対応化が始まっています）

- partedコマンドでGPTまたはMBRパーティションを作成、設定変更、削除できます。

parted コマンド

```
parted [オプション] [デバイス [操作コマンド [操作オプション...]...]]
```

※操作コマンドを省略すると、対話モードに入ります。

オプション	説明
-l	パーティション一覧を表示

操作コマンド	説明
print	パーティション一覧を表示
mkpart	パーティションを作成
rm	パーティションを削除
quit	partedコマンドを終了
help	操作コマンドのヘルプを表示

パーティションの確認

```
# parted -l
```

```
...
```

```
モデル: JMicron Tech (scsi)
```

```
ディスク /dev/sdb: 256GB
```

```
セクタサイズ (論理/物理): 512B/4096B
```

```
パーティションテーブル: gpt
```

```
ディスクフラグ:
```

番号	開始	終了	サイズ	ファイルシステム	名前	フラグ
1	1049kB	211MB	210MB	fat16	EFI System Partition	boot
2	211MB	1285MB	1074MB	xf		
3	1285MB	63.3GB	62.0GB			lvm
...						

パーティションの作成

```
# parted /dev/sdb
GNU Parted 3.1
/dev/sdb を使用
GNU Parted へようこそ！ コマンド一覧を見るには 'help' と入力してください。
(parted) print
モデル: JMicron Tech (scsi)
ディスク /dev/sdb: 256GB
セクタサイズ (論理/物理): 512B/4096B
パーティションテーブル: gpt
ディスクフラグ:
```

番号	開始	終了	サイズ	ファイルシステム	名前	フラグ
1	1049kB	211MB	210MB	fat16	EFI System Partition	boot
2	211MB	1285MB	1074MB	xf		
3	1285MB	63.3GB	62.0GB			lvm

パーティションの作成

```
(parted) mkpart part4 63.3G 140G ※ 「part4」 は任意の名前です
```

```
(parted) print
```

```
...
```

番号	開始	終了	サイズ	ファイルシステム	名前	フラグ
1	1049kB	211MB	210MB	fat16	EFI System Partition	boot
2	211MB	1285MB	1074MB	xf		
3	1285MB	63.3GB	62.0GB			lvm
4	63.3GB	140GB	76.7GB		part4	

```
(parted) mkpart part5 140G -0 ※ 「-0」 はディスクの最後を指定しています
```

```
(parted) print
```

```
...
```

番号	開始	終了	サイズ	ファイルシステム	名前	フラグ
1	1049kB	211MB	210MB	fat16	EFI System Partition	boot
2	211MB	1285MB	1074MB	xf		
3	1285MB	63.3GB	62.0GB			lvm
4	63.3GB	140GB	76.7GB		part4	
5	140GB	256GB	116GB		part5	

パーティションの削除、partedコマンド終了

```
(parted) rm 5
```

```
(parted) print
```

```
...
```

番号	開始	終了	サイズ	ファイルシステム	名前	フラグ
1	1049kB	211MB	210MB	fat16	EFI System Partition	boot
2	211MB	1285MB	1074MB	xf		
3	1285MB	63.3GB	62.0GB			lvm
4	63.3GB	140GB	76.7GB		part4	

```
(parted) quit
```

通知: 必要であれば /etc/fstab を更新するのを忘れないようにしてください。

- gdiskコマンドでも、GPTまたはMBRパーティションを作成、設定変更、削除できます。
- fdiskコマンドの使い方も同様です。但し、MBR専用とされています。

gdisk コマンド

```
gdisk [オプション] デバイス
```

オプション	説明
-l	パーティション一覧を表示

※以降、対話モードで操作します。

操作コマンド	説明
p	パーティション一覧を表示
n	パーティションを作成
d	パーティションを削除
t	パーティションタイプを変更
q	変更を保存せずに終了
w	変更を保存して終了
m	操作コマンドのヘルプを表示

パーティションの確認

```
# gdisk /dev/sdb
GPT fdisk (gdisk) version 0.8.10

Partition table scan:
  MBR: protective
  BSD: not present
  APM: not present
  GPT: present

Found valid GPT with protective MBR; using GPT.

Command (? for help): p
Disk /dev/sdb: 500118192 sectors, 238.5 GiB
. . .
Number  Start (sector)    End (sector)  Size      Code  Name
   1            2048             411647      200.0 MiB  EF00  EFI System Partition
   2           411648           2508799     1024.0 MiB  0700
   3          2508800          123629567    57.8 GiB   8E00
   4          123629568          273436671    71.4 GiB   0700  part4
```

パーティションの作成

```
Command (? for help): n
Partition number (5-128, default 5): (ENTER)
First sector (34-500118158, default = 273436672) or {+-}size{KMGTP}: (ENTER)
Last sector (273436672-500118158, default = 500118158) or {+-}size{KMGTP}: +60G
Current type is 'Linux filesystem'
Hex code or GUID (L to show codes, Enter = 8300): (ENTER)
Changed type of partition to 'Linux filesystem'
```

```
Command (? for help): p
```

. . .

Number	Start (sector)	End (sector)	Size	Code	Name
1	2048	411647	200.0 MiB	EF00	EFI System Partition
2	411648	2508799	1024.0 MiB	0700	
3	2508800	123629567	57.8 GiB	8E00	
4	123629568	273436671	71.4 GiB	0700	part4
5	273436672	399265791	60.0 GiB	8300	Linux filesystem

パーティションの作成

```
Command (? for help): n
Partition number (6-128, default 6): (ENTER)
First sector (34-500118158, default = 399265792) or {+-}size{KMGTP}: (ENTER)
Last sector (399265792-500118158, default = 500118158) or {+-}size{KMGTP}: (ENTER)
Current type is 'Linux filesystem'
Hex code or GUID (L to show codes, Enter = 8300): (ENTER)
Changed type of partition to 'Linux filesystem'
```

```
Command (? for help): p
```

. . .

Number	Start (sector)	End (sector)	Size	Code	Name
1	2048	411647	200.0 MiB	EF00	EFI System Partition
2	411648	2508799	1024.0 MiB	0700	
3	2508800	123629567	57.8 GiB	8E00	
4	123629568	273436671	71.4 GiB	0700	part4
5	273436672	399265791	60.0 GiB	8300	Linux filesystem
6	399265792	500118158	48.1 GiB	8300	Linux filesystem

パーティションの削除、変更の保存とgdiskコマンド終了

Command (? for help): d

Partition number (1-6): 6

Command (? for help): p

. . .

Number	Start (sector)	End (sector)	Size	Code	Name
1	2048	411647	200.0 MiB	EF00	EFI System Partition
2	411648	2508799	1024.0 MiB	0700	
3	2508800	123629567	57.8 GiB	8E00	
4	123629568	273436671	71.4 GiB	0700	part4
5	273436672	399265791	60.0 GiB	8300	Linux filesystem

Command (? for help): w ※変更を保存して終了。「q」で終了すると、変更が破棄されます。

Final checks complete. About to write GPT data. THIS WILL OVERWRITE EXISTING PARTITIONS!!

Do you want to proceed? (Y/N): y

. . .

The operation has completed successfully.

fdiskコマンドのGPT対応化は「実験フェーズ」

```
# fdisk /dev/sdb
```

```
WARNING: fdisk GPT support is currently new, and therefore in an experimental phase. Use
at your own discretion.
```

```
Welcome to fdisk (util-linux 2.23.2).
```

```
...
```

```
コマンド (m でヘルプ): p
```

```
...
```

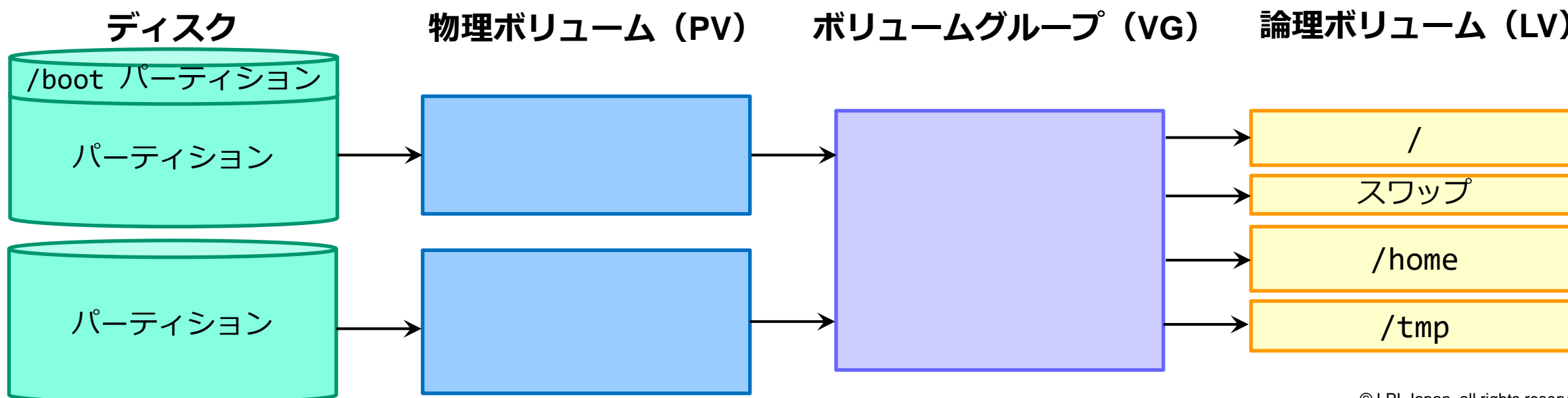
#	Start	End	Size	Type	Name
1	2048	411647	200M	EFI System	EFI System Partition
2	411648	2508799	1G	Microsoft basic	
3	2508800	123629567	57.8G	Linux LVM	
4	123629568	273436671	71.4G	Microsoft basic part4	
5	273436672	399265791	60G	Linux filesystem	Linux filesystem

```
コマンド (m でヘルプ): q
```

- 1.はじめに
- 2.パーティションの概要
- 3.パーティション分割とディレクトリ構成
- 4.デバイスファイル
- 5.パーティションの作成
- 6.LVMの概要**

- partedコマンドとgdiskコマンドでパーティションを作成しましたが、これらは作成した時点でサイズが決まっています。作成後に拡張や縮小することはできません。
- LVM (Logical Volume Manager 論理ボリュームマネージャ) は、複数のディスクやパーティションにまたがった論理的なパーティションである「論理ボリューム」を構成することができる仕組みです。論理ボリュームは、作成後でも拡張や縮小が容易に可能です。
- LinuCレベル1では、LVMに関する基礎知識が出題されます。LVMの操作は、レベル2の出題範囲です。

- ディスクやパーティションをLVMで扱えるようにしたものを、物理ボリューム（PV：Physical Volume）と呼びます。
- 物理ボリュームを束ねて塊とし、ボリュームグループ（VG）を構成します。論理的な大容量ディスクを作ることができます。
- ボリュームグループから必要なサイズを切り出して、論理的なパーティションを作成します。これを論理ボリューム（LV）と呼びます。物理的パーティションと同様に、論理ボリューム上にファイルシステムを作成することができます。（ブートローダはLVを認識しないので、/boot は物理的パーティション上に構成します）



LVMの確認

```
# parted -l
```

```
. . .
```

```
モデル: JMicron Tech (scsi)
```

```
ディスク /dev/sdb: 256GB
```

```
セクタサイズ (論理/物理): 512B/4096B
```

```
パーティションテーブル: gpt
```

```
ディスクフラグ:
```

番号	開始	終了	サイズ	ファイルシステム	名前	フラグ
1	1049kB	211MB	210MB	fat16	EFI System Partition	boot
2	211MB	1285MB	1074MB	xf		※ /boot
3	1285MB	63.3GB	62.0GB			lvm ※ PV
4	63.3GB	140GB	76.7GB		part4	
5	140GB	204GB	64.4GB		Linux filesystem	

LVMの確認

```
モデル: Linux device-mapper (linear) (dm)
ディスク /dev/mapper/centos_ssd1--centos7-swap: 8321MB
セクタサイズ (論理/物理): 512B/4096B
パーティションテーブル: loop
ディスクフラグ:
```

番号	開始	終了	サイズ	ファイルシステム	フラグ
1	0.00B	8321MB	8321MB	linux-swap(v1)	※ LV (スワップパーティション)

```
モデル: Linux device-mapper (linear) (dm)
ディスク /dev/mapper/centos_ssd1--centos7-root: 53.7GB
セクタサイズ (論理/物理): 512B/4096B
パーティションテーブル: loop
ディスクフラグ:
```

番号	開始	終了	サイズ	ファイルシステム	フラグ
1	0.00B	53.7GB	53.7GB	xf	※ LV (ルートパーティション)

LVMの確認

```
# gdisk /dev/sdb
GPT fdisk (gdisk) version 0.8.10
. . .
Number  Start (sector)    End (sector)  Size        Code  Name
   1            2048             411647       200.0 MiB   EF00  EFI System Partition
   2           411648             2508799      1024.0 MiB   0700
   3           2508800          123629567    57.8 GiB    8E00
   4          123629568          273436671    71.4 GiB    0700  part4
   5          273436672          399265791    60.0 GiB    8300  Linux filesystem
```

```
Command (? for help): l      ※パーティションタイプの一覧を表示する
0700 Microsoft basic data    0c01 Microsoft reserved      2700 Windows RE
3000 ONIE boot                3001 ONIE config              4100 PowerPC PReP boot
4200 Windows LDM data         4201 Windows LDM metadata     7501 IBM GPFS
7f00 ChromeOS kernel         7f01 ChromeOS root            7f02 ChromeOS reserved
8200 Linux swap               8300 Linux filesystem          8301 Linux reserved
8302 Linux /home              8400 Intel Rapid Start        8e00 Linux LVM
. . .
```

```
Command (? for help): q
```


LVMの確認

```
# df -h ※マウントされているファイルシステムを表示する (dfコマンドは副題1.05.3の出題範囲)
ファイルシス      サイズ  使用  残り  使用% マウント位置
devtmpfs           3.8G    0  3.8G    0% /dev
tmpfs              3.8G    0  3.8G    0% /dev/shm
tmpfs              3.8G  10M  3.8G    1% /run
tmpfs              3.8G    0  3.8G    0% /sys/fs/cgroup
/dev/mapper/centos_ssd1--centos7-root  50G  4.6G  46G   10% /
/dev/sdb2          1014M  172M  843M   17% /boot
/dev/sdb1          200M   12M  189M    6% /boot/efi ※ ESP
tmpfs              774M   24K  774M    1% /run/user/1000
```

1. CentOS 7 に基づく学習環境を自分で構築してみましょう。
2. パーティションの規格には、MBR と GPT とがあります。
3. FHSに準拠したディレクトリ名称を使うことが推奨されています。ディスク上には、少なくともルートパーティションとスワップパーティションの2つが必要です。
4. パーティションのデバイスファイル名は、`/dev/sda1`、`/dev/sda2`・・・などとなります。
5. パーティションの作成、設定変更、削除には、`parted`コマンド、`gdisk`コマンド、`fdisk`コマンドを使います。`parted`と`gdisk`はMBRとGPTの両用、`fdisk`はMBR専用です。
6. LVMを使って作成される論理ボリュームは、作成後でも拡張や縮小が容易に可能です。

ご清聴ありがとうございました