

LinuC レベル 2 Version10.0

技術解説無料セミナー

2020/12/5 開催

主題 2.02
ファイルシステムとストレージ管理

本日の講師



エスディーテック株式会社
末永 貴一

■自己紹介



エスディーテック株式会社

– 組み込み機器向けUI/UXソフトウェアの研究開発

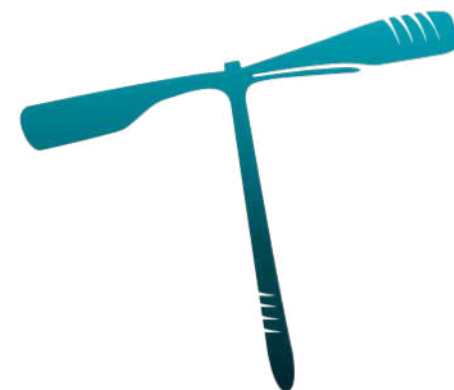
<http://www.sdtech.co.jp>

Linux関連文章の執筆

- LPIC Level2 1回で合格必達テキスト+問題集
- LPI-Japan コラム【Linux道場 入門編】
- @IT 「Linuxをいまから学ぶコツ教えます」
- @IT 「Linuxに触れよう」
- 日経Linux 「Xと次世代「Wayland」を知る」
など



#LinuC学習中





■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

■ Version 10.0と従来の出題範囲の比較



#LinuC学習中

テーマ	Version 10.0	従来
LinuC-1	仮想技術 <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 	← (Version 10.0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 	← (Version 10.0で新設)
	その他 <p style="text-align: center;">→ (Version 10.0で削除)</p>	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-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 	← (Version 10.0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 	← (Version 10.0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) 	← (Version 10.0で出題範囲に含む)
	その他 <p style="text-align: center;">→ (Version 10.0で削除)</p>	RAID、記憶装置へのアクセス方、FTPサーバーの保護、他



今回のテーマ

主題2.02 : ファイルシステムとストレージ管理

- ✓ ファイルシステムとは
- ✓ VFS
- ✓ i-node
- ✓ マウント
- ✓ ファイルシステム管理
- ✓ LVM

■ファイルシステムとは



#LinuC学習中

✓物理的な記憶装置に保存されたデータを管理する機能

- 基本的にはOSの機能として提供されるが、OS自体の機能とは分離されている
- OSがデータを効率よく管理するために様々な種類のファイルシステムがある

✓Linuxで利用されるファイルシステム

- デフォルトではExt4
- その他にXFS、Btrfs、vfat等複数のファイルシステムをサポート
- 用途によってファイルシステムを選択する

■VFS (Virtual File System) とは



#LinuC学習中

✓ファイルシステム上に構築された抽象化レイヤー

- 統一的なファイルアクセスという入出カインターフェイスを提供
- データ、デバイスを含む異なるコンピュータリソースを意識しない
- ファイルシステムが違ってても同一の方法でアクセス可能

■ VFSのアクセスイメージ



#LinuC学習中

ファイル操作のコマンド
ls,cd,cp,等

同一の方法で異なるターゲット
に対してアクセスできる



VFS

異なるファイル
システム群



EXT4

XFS

Btrfs

VFAT

NTFS



デバイスの相違を
意識しない

■ i-node (index-node) とは

✓ ファイルを管理するためのデータ構造体

- VFSで抽象化されたデータと実データを関連づけるデータ
 - ファイルシステムごとに作成される
 - i-nodeに含まれる主なデータ
 - i-node番号
 - デバイスID
 - ファイルモード
 - ファイルのUID
 - ファイルのGID
 - アクセス情報
 - リンク数
- 等



#LinuC学習中

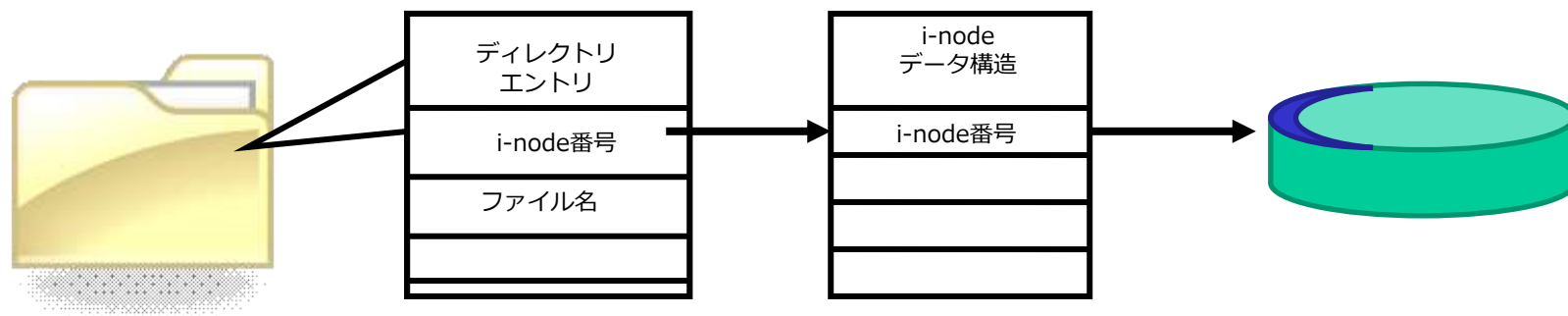
➤ i-node情報の確認はstatコマンドで行う

例：

```
$ stat -L ファイル名
```

■ファイルアクセス

 #LinuC学習中



ファイル操作の内部動作

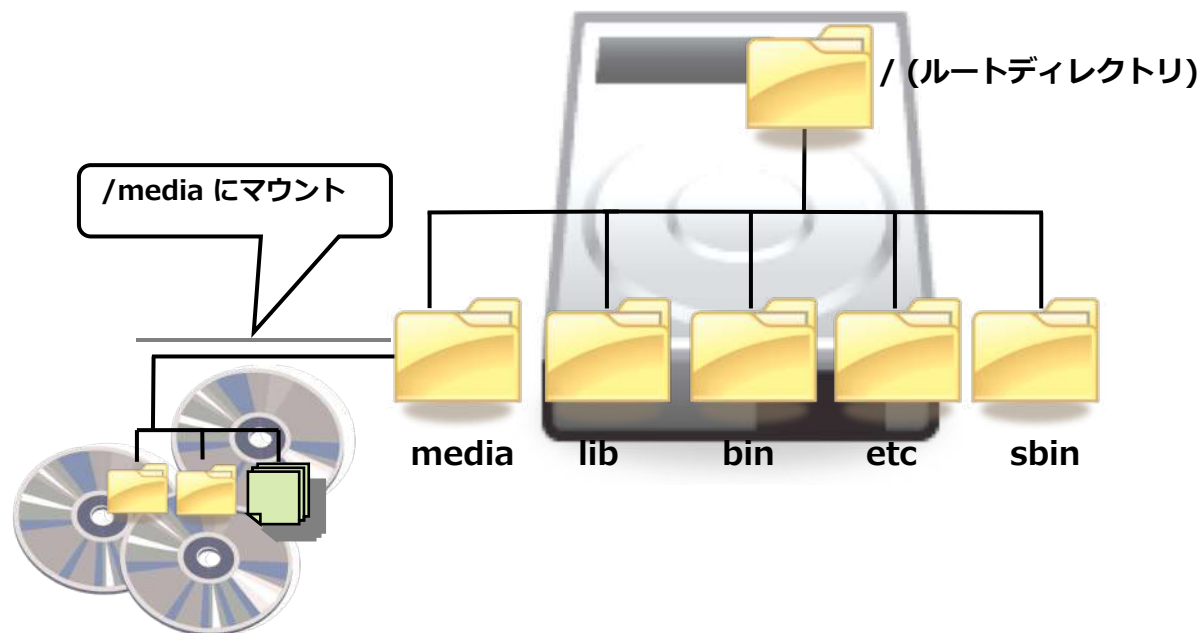
■マウントとは



#LinuC学習中

✓異なるファイルシステム（デバイス）を使用するための操作

- ハードウェアが認識されてもそのままだと利用できない
- ルートファイルシステムに他のファイルシステムを接ぎ木して、ファイル操作を可能にする
- マウントしたファイルシステムはアンマウント（S/W的接続解除）し、物理的接続解除
 - マウントしたまま物理的接続解除を行うとデータを破損する可能性がある



■マウント系のコマンド



#LinuC学習中

✓ 明示的マウント

mount、umountを実行

```
ex. mount -t vfat /dev/cdrom /media
    umount /media
```

※ マウントしたデバイスを取り出すときにはumountを実行し、マウント解除する

✓ 定義済みマウント

ファイルシステムテーブル/etc/fstabに記述
起動時に自動的にマウントを行うことが可能

- /etc/fstabファイルの書式

```
fs_spec fs_file fs_vfstype fs_mntops fs_freq fs_passno
```

■fstabの書式

fs_spec fs_file fs_vfstype fs_mntops fs_freq fs_passno



#LinuC学習中

No.	フィールド名	意味
1	fs_spec (デバイスファイル)	マウントしたいデバイスのファイルを指定
2	fs_file (マウントポイント)	ファイルシステムのどこにマウントするかを指定
3	fs_vfstype (ファイルシステム)	マウントするファイルシステムのタイプを指定
4	fs_mntops (オプション)	マウントする際のオプションを指定
5	fs_freq (ダンプフラグ)	dumpコマンドでバックアップの対象とするか： 0 (しない) 0以外 (n日ごとにdumpする予定)
6	fs_pasno (fsckチェックフラグ)	fsckを利用するか： 0 (利用しない) 1 (ルートファイルシステムで使用) 2 (その他のファイルシステムで使用)

■ファイルシステムの保守

✓ ファイルシステムが適切に運用されているかを管理

- ストレージの状態確認
- 残容量のチェック
- ファイル数のチェック
- ファイルシステム破損時の対応

✓ ファイルシステム操作コマンド

- mkfs : ファイルシステム作成
- fsck : ファイルシステムチェック
- tune2fs : ファイルシステムのパラメータ操作
- dumpe2fs : ファイルシステムの情報表示
- df : ファイルシステムディスク情報の表示



#LinuC学習中

■Ext系ファイルシステムの基本構造（概要）

- ✓ 複数のブロックをブロックグループにまとめて管理する
 - スーパーブロック：ファイルシステムに関する管理情報
 - 管理情報データブロック：i-node等のファイルの管理情報
 - データブロック：実際のデータ



#LinuC学習中



■ファイルシステムの操作例



#LinuC学習中

- ✓ ファイルシステムのチェック

```
# fsck -t ext4 /dev/sda1
```

- ✓ ファイルシステムの修復

```
# fsck -b 32768 /dev/sda1
```

- ✓ 不良ブロックの調査

```
# fsck -c /dev/sda1
```

- ✓ HDD,SSDの自己診断機能

```
# smartctl --all /dev/sda1
```


■LVM(Logical Volume Manager)とは



#LinuC学習中

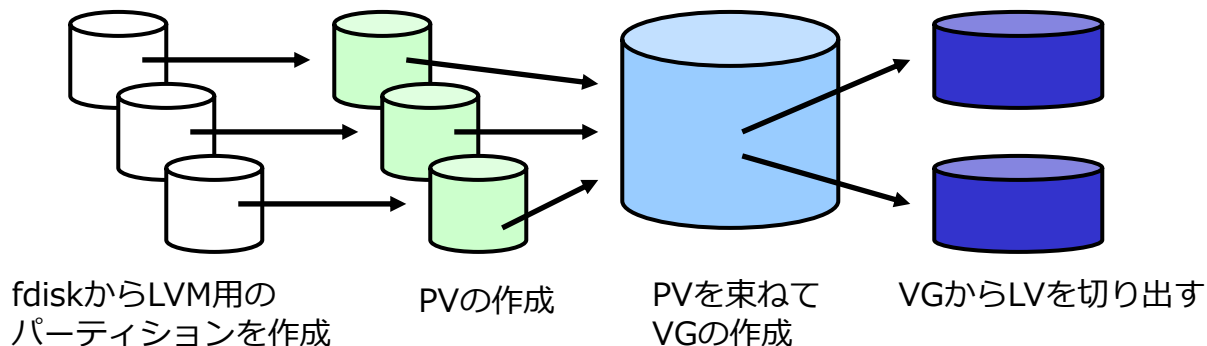
- ✓ 複数の異なるパーティション、物理ストレージを論理的に束ねるシステム
 - 動的なサイズ変更が可能
 - 物理的に異なるストレージ領域を1つの領域として扱うことができる

✓ LVMの構成要素

PV(Physical Volume) : LVM用に構成された、物理的なボリューム

VG(Volume Group) : 1つ以上のPVを束ねて構成される、論理的なボリューム

LV(Logical Volume) : VGから実際に使用する領域を切り出した論理的ボリューム



■ LVMのコマンド



#LinuC学習中

✓ PVの作成

- # pvcreate [PVにするデバイス ...]
- Ex. # pvcreate /dev/sdc1 /dev/sdc2 /dev/sdc3

✓ VGの作成

- # vgcreate [VG名] [VGに加えるPVデバイス]
- Ex. # vgcreate sample_vg /dev/sdc1 /dev/sdc2 /dev/sdc3

✓ LVの作成

- # lvcreate -n[LV名] -L[サイズ] [VG名]
- Ex. # lvcreate -n sample_lv -L 3g sample_vg

■まとめ

- ✓ ファイルシステム
 - データを効率よく管理す機能
- ✓ VFS
 - ファイルシステム上に構築された統一的なファイルアクセスインターフェイス
- ✓ i-node
 - ファイル管理のためのデータ構造体
- ✓ マウント
 - 異なるファイルシステムをファイル操作可能にするための操作
- ✓ ファイルシステム管理
 - ファイルシステム（ストレージ）の状態を把握し、適切な処理をする
 - ファイルシステム操作コマンド
 - Ext系ファイルシステムの基本構造
- ✓ LVM
 - 複数の物理ストレージ領域を統合し1つの領域として扱える



#LinuC学習中



Q & A



ありがとうございました



<https://www.sdtech.co.jp>