

LinuC レベル1 Version10.0 技術解説無料セミナー

2021/3/6 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	1.05.3 ファイルシステムの作成と管理、マウント

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。
- 2021年1月23日、[LinuCレベル1 Version10.0 技術解説セミナー「ハードディスクのレイアウトとパーティション」](#)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

ファイルシステムの作成と管理、マウント

- 1.はじめに
- 2.ディスク装置とファイルシステム
- 3.ファイルシステムの概要
- 4.ファイルシステムの作成
- 5.ファイルシステムのマウントと管理

1.はじめに

2.ディスク装置とファイルシステム

3.ファイルシステムの概要

4.ファイルシステムの作成

5.ファイルシステムのマウントと管理

- 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、ファイルシステムに関する知識と、ファイルシステムを作成・マウント・管理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、ディスクに予めパーティションを作成しておくことが必要です。2021年1月23日の[LinuCレベル1技術解説セミナー「ハードディスクのレイアウトとパーティション」](#)を参照してください。

1.はじめに

2.ディスク装置とファイルシステム

3.ファイルシステムの概要

4.ファイルシステムの作成

5.ファイルシステムのマウントと管理

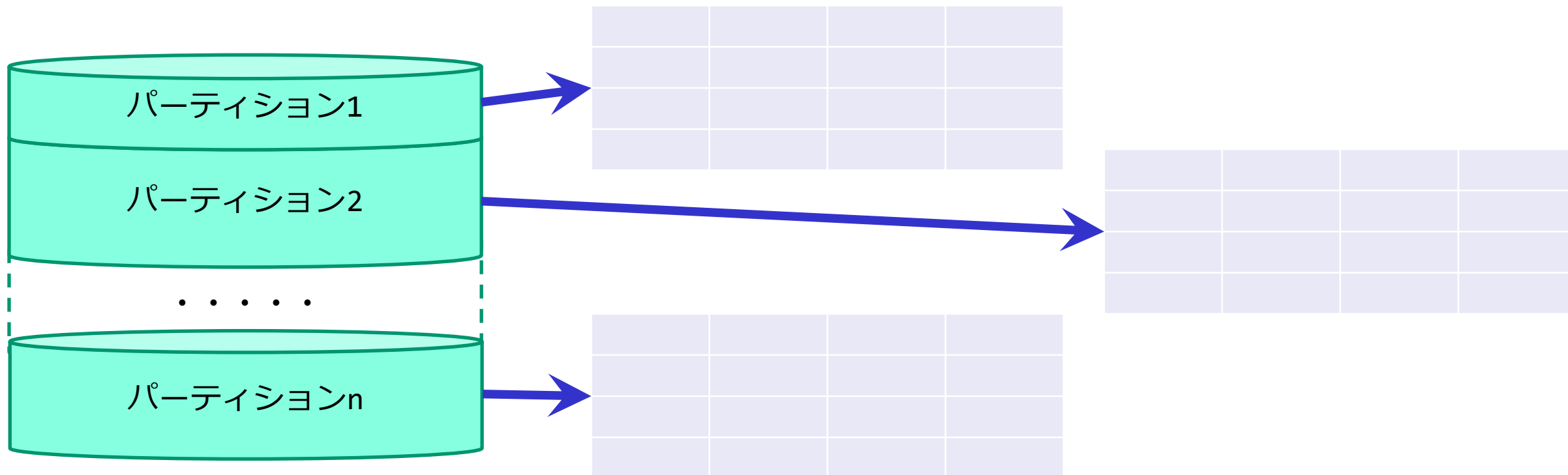
- Linuxシステムにディスク装置を接続して、利用できるようにするには、パーティションに分割し、パーティション毎にファイルシステムを作成し、ルートファイルシステムにマウントする必要があります。



- パーティション（副題 1.05.2）については、1月23日の [LinuCレベル1技術解説セミナー](#) で説明しました。
- 本日は、ファイルシステムの作成・マウント・管理（副題 1.05.3）について学習します。

- 1.はじめに
- 2.ディスク装置とファイルシステム
- 3.ファイルシステムの概要**
- 4.ファイルシステムの作成
- 5.ファイルシステムのマウントと管理

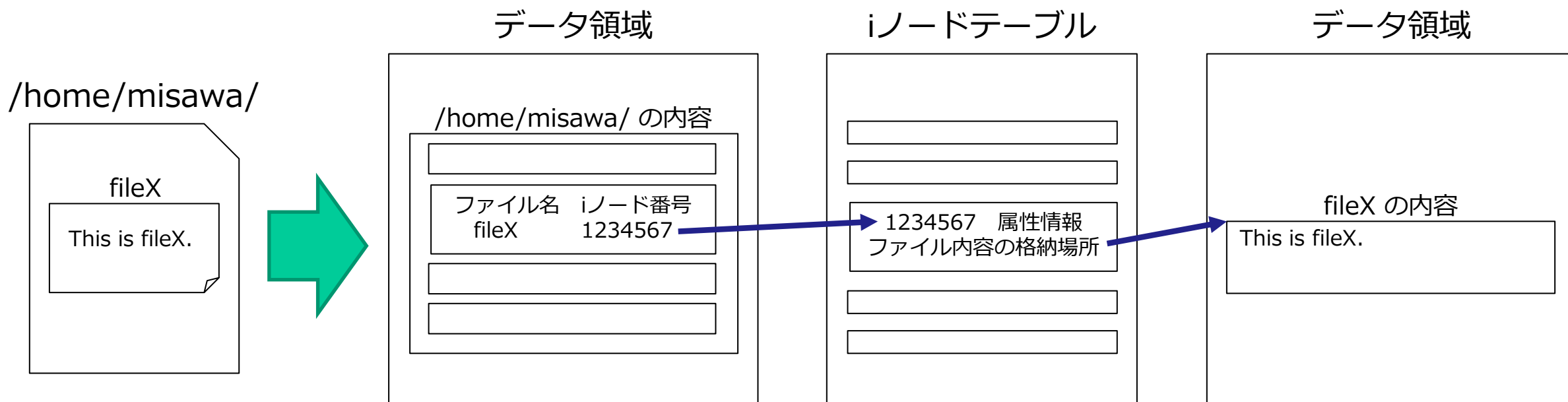
- ディスクに新しいパーティションを作成する方法を、1月23日のセミナーで説明しました。しかし、パーティションを作っただけでは、そこにファイルを格納することはできません。パーティションを利用可能にするには、そこにファイルシステムを作成します。「フォーマットする」と言われることもあります。
- ファイルシステムはディスクにファイルを格納する仕組みで、パーティションまたは論理ボリュームごとに作成します。



- ファイルシステムには様々な種類があり（後述）、それぞれ仕組みが異なりますが、多くは以下のような要素から構成されています。

構成要素	説明
全体管理情報領域	ファイルシステム全体を管理する情報を格納。 ファイルシステムの種類やサイズ、ラベル、マウント情報等。
iノードテーブル	各ファイルを管理するiノードの情報を格納。 ファイルの属性に関する情報、ファイル内容の格納場所。
データ領域	ファイル内容のデータを格納。

- iノードは、ファイル毎の管理情報です。iノード番号、ファイルの属性情報（ls -l で表示されるファイルの所有者、グループ、パーミッション、作成日付など）、ファイル内容を格納している場所の情報（ポインタ）が管理されています。
- ファイル内容はデータ領域に格納され、その格納場所を iノードが管理します。
- ファイル名は iノードではなく、そのファイルを含んでいるディレクトリのファイル内容として管理され、iノード番号に紐付けられます。



- Linuxで取り扱うことのできる主なファイルシステムには、以下のようなものがあります。

ファイルシステムタイプ	特徴
XFS	RHEL/CentOS7/8の標準ファイルシステム。 内部を複数のアロケーショングループに分割し、並行処理することで、I/Oを高速化している。
ext4	ext3 を更に改良し、性能や信頼性等が改善された。 Debian10の標準ファイルシステム。
ext3	ext2 を改良し、ジャーナリング機能が追加された。CentOS6 の標準ファイルシステム。
ext2	初期のLinuxに標準として採用されたファイルシステム。
Btrfs	XFSより新しいファイルシステム。高度な機能を持つ。
vfat	Microsoft Windows と Linux の両方で利用できるファイルシステム。
iso9660	データCD用のファイルシステム。
tmpfs	メモリ上に生成するファイルシステム。

- 1.はじめに
- 2.ディスク装置とファイルシステム
- 3.ファイルシステムの概要
- 4.ファイルシステムの作成**
- 5.ファイルシステムのマウントと管理

- mkfsコマンドでファイルシステムを作成することができます。

mkfs コマンド

mkfs [オプション] デバイス

オプション

説明

-t ファイルシステムタイプ	作成するファイルシステムの種類を指定する
-L ラベル	ラベル名を付与する

- mkfsコマンドは、-t オプションで指定されたファイルシステムタイプ毎の作成コマンドを呼び出して、実行します。例えば「mkfs -t xfs」とすると、mkfs.xfsコマンドが実行され、XFSファイルシステムが作成されます。この場合、直接に mkfs.xfsコマンドを実行しても同じです。
- ext2/3/4ファイルシステムを作成するには、mke2fsコマンドも使用できます。-t でタイプを指定しますが、デフォルトは ext2 です。

- findfs コマンドでファイルシステムを検索できます。

findfs コマンド

```
findfs [LABEL=ラベル名 | UUID=uuid]
```

- xfs_admin コマンドは、XFSファイルシステムのラベル名やUUIDを表示・設定します。

xfs_admin コマンド

```
xfs_admin [オプション] デバイス
```

オプション	説明
-l	ラベルを表示する
-u	UUIDを表示する
-L ラベル名	ラベル名を設定する
-U UUID	UUIDを設定する

ext2/3/4ファイルシステムの場合は、tune2fsコマンドを使用します。

ファイルシステムの作成

```
# 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
4	63.3GB	140GB	76.7GB		part4	
5	140GB	204GB	64.4GB		Linux filesystem	

```
. . .
```

ファイルシステムが作成されていない /dev/sdb4と5 に作成します。

ファイルシステムの作成

```
# mkfs -t xfs /dev/sdb4      ※mkfs.xfs /dev/sdb4 でも同様
meta-data=/dev/sdb4        isize=512    agcount=4, agsize=4681472 blks
    =                       sectsz=4096  attr=2, projid32bit=1
    =                       crc=1        finobt=0, sparse=0
data      =                 bsize=4096  blocks=18725888, imaxpct=25
    =                       sunit=0        swidth=0 blks
naming    =version 2        bsize=4096  ascii-ci=0 ftype=1
log       =internal log    bsize=4096  blocks=9143, version=2
    =                       sectsz=4096  sunit=1 blks, lazy-count=1
realtime  =none            extsz=4096  blocks=0, rtextents=0

# xfs_admin -L p4fs /dev/sdb4  ※mkfsコマンドの-Lオプションでもラベル名の付与が可能
writing all SBs
new label = "p4fs"

# findfs LABEL=p4fs
/dev/sdb4
```

ファイルシステムの作成

```
# mke2fs -t ext4 /dev/sdb5      ※mkfs -t ext4 /dev/sdb5 でも同様
```

```
mke2fs 1.42.9 (28-Dec-2013)
```

```
. . .
```

```
# 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
4	63.3GB	140GB	76.7GB	xf	part4	
5	140GB	204GB	64.4GB	ext4	Linux filesystem	
. . .						

- ファイルシステム以外に、スワップ領域が必要です。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、処理を続けます。
- スワップ領域は一般にシステムインストール時に生成しますが、追加する場合は、mkswapコマンドで空のパーティションに作成します。

mkswap コマンド

```
mkswap [オプション] デバイス
```

- 作成したスワップ領域を有効化するには、swaponコマンドを使用します。

swapon コマンド

```
swapon [オプション] デバイス
```

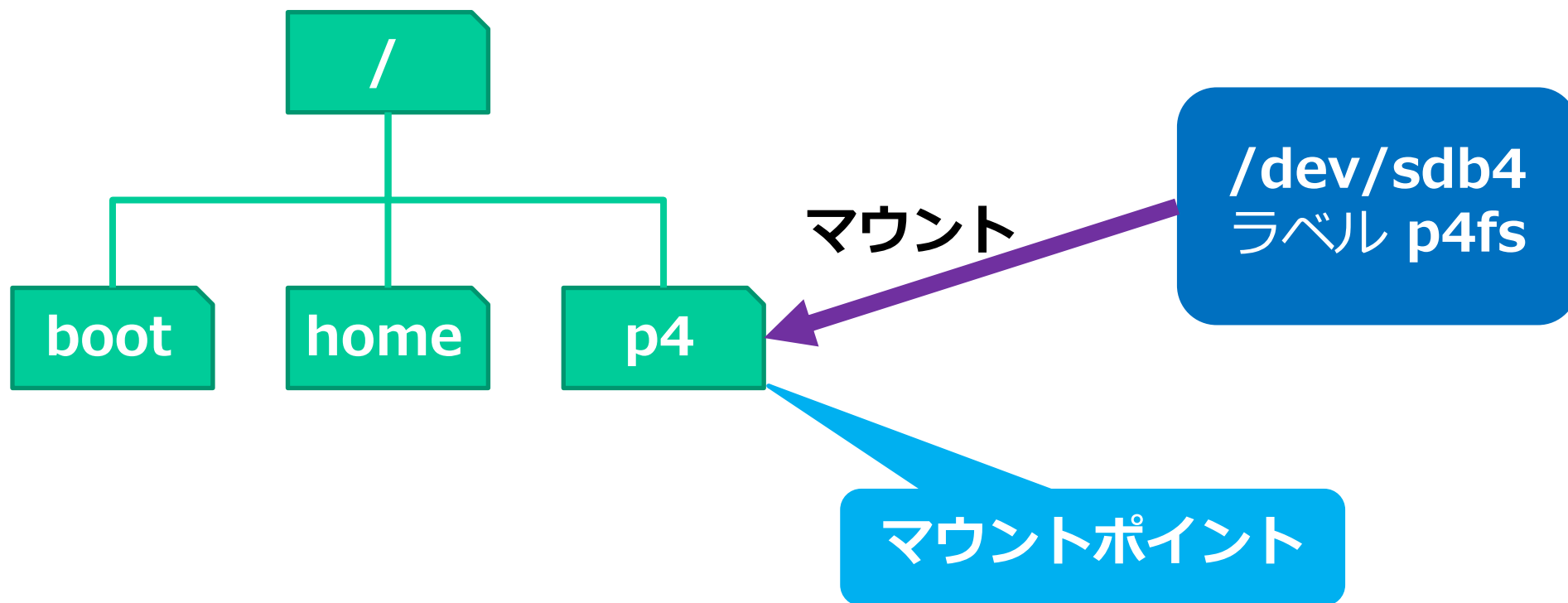
- スワップ領域を無効化するには、swapoffコマンドを使用します。

swapoff コマンド

```
swapoff [オプション] デバイス
```

- 1.はじめに
- 2.ディスク装置とファイルシステム
- 3.ファイルシステムの概要
- 4.ファイルシステムの作成
- 5.ファイルシステムのマウントと管理**

- 新しく作成したファイルシステムはディレクトリに紐づけられていないため、まだファイルを格納することができません。これらのファイルシステムをルートファイルシステムの階層構造に組み込むことにより、利用できるようにします。この操作を「マウント」といいます。
- ルートファイルシステム内のマウントされる場所を「マウントポイント」と呼びます。



- mountコマンドは、マウント状況の表示と、指定したファイルシステムのマウントを行います。

マウント状況を表示する

```
mount [-l]
```

ファイルシステムをマウントする

```
mount [-t ファイルシステムタイプ] [-o オプション] デバイス マウントポイント
```

/etc/fstabファイル（後述）に記述されたファイルシステムをマウントする

```
mount -a
```

- マウントを解除することを「アンマウントする」といいます。umountコマンドを使用します。

ファイルシステムをアンマウントする

```
umount [マウントポイント | デバイス]
```

ファイルシステムのマウント

```

# mount
sysfs on /sys type sysfs (rw,nosuid,nodev,noexec,relatime,seclabel)
. . .
/dev/mapper/centos-root on / type xfs (rw,relatime,seclabel,attr2,inode64,noquota)
. . .
/dev/sdb2 on /boot type xfs (rw,relatime,seclabel,attr2,inode64,noquota)
/dev/sdb1 on /boot/efi type vfat (rw,relatime,fmask=0077,dmask=0077,codepage=437,ioccharse
. . .
# mkdir /p4                ※マウントポイントを作成する
# mount -t xfs /dev/sdb4 /p4  ※マウントする
# mount
. . .
tmpfs on /run/user/1001 type tmpfs (rw,nosuid,nodev,relatime,seclabel,size=39454
/dev/sdb4 on /p4 type xfs (rw,relatime,seclabel,attr2,inode64,noquota)    ※マウントされた

# cd /p4
# touch fileX             ※マウントしたファイルシステムにファイルを作成する。
# ls
fileX

```

ファイルシステムのマウント

```

# umount /p4
umount: /p4: target is busy.      ※使用中のファイルシステムはアンマウントできない
(In some cases useful info about processes that use
the device is found by lsof(8) or fuser(1))

# cd
# umount /p4      ※使用していない状態でアンマウントする、マウントポイントを指定した実行例
# ls /p4
    ※アンマウントしたので、作成したファイル fileX にアクセスできない。
# mount
. . .
tmpfs on /run/user/0 type tmpfs (rw,nosuid,nodev,relatime,seclabel,size=792148k,mode=700)
    ※/dev/sdb4 がアンマウントされた。

# mkdir /p5
# mount -t ext4 /dev/sdb5 /p5
# mount
. . .
/dev/sdb5 on /p5 type ext4 (rw,relatime,seclabel,data=ordered)      ※マウントされた

```

- mountコマンドを使ってファイルシステムをマウントしましたが、システムが再起動すると、マウントは外れてしまいます。システム起動時にファイルシステムが自動的にマウントされるようにするには、/etc/fstabファイルに設定を記述します。

/etc/fstabファイルの書式

デバイス マウントポイント ファイルシステムタイプ オプション ダンプ チェック

※フィールド区切りは TAB または 空白文字

フィールド	説明
デバイス	デバイス名、ラベル名、UUID のいずれかで指定する
マウントポイント	マウントポイントを指定する
ファイルシステムタイプ	ファイルシステムの種類を指定する
オプション	次スライド参照
ダンプ	バックアップコマンドdumpが参照するフラグ 0：バックアップしない 1：バックアップする
チェック	起動時のファイルシステムチェックの優先順位 0：チェックしない 1：最優先 2以上：小さい値ほど優先

主なオプション	説明
defaults	デフォルトのオプションセット (auto, exec, rw, nouserなど)
auto/noauto	auto : 自動マウントを行う noauto : 自動マウントを行わない
exec/noexec	exec : プログラムなどの実行を許可する noexec : 実行を許可しない
ro/rw	ro : 読み取り専用 rw : 読み書き両用
nouser	スーパーユーザだけがマウントできる
user	一般ユーザのマウント操作を許可する、アンマウントはマウントしたユーザのみ
users	一般ユーザのマウント操作を許可する、アンマウントは誰でも可能

自動マウントの設定

```
# xfs_admin -l /dev/sdb4
label = "p4fs"

# vi /etc/fstab
. . .
/dev/mapper/centos-root      /                xfs      defaults        0 0
UUID=d0e0491f-e162-4d8e-b2f8-658ab8f9e9b9  /boot           xfs      defaults        0 0
UUID=666A-E57F              /boot/efi       vfat     umask=0077,shortname=winnt  0 0
. . .
LABEL=p4fs                  /p4              xfs      defaults        1 2      ※エントリを追加
/dev/sdb5                   /p5              ext4     defaults        1 2      ※エントリを追加
```

※デバイスは、デバイス名、ラベル名、UUID のいずれかで指定します。

- /etc/fstabファイルに記述されているファイルシステムは、mountコマンドの引数にマウントポイントのみを指定してマウントできます。

/etc/fstabファイルに記述されているファイルシステムのマウント

```
# mount
. . .
/dev/sdb5 on /p5 type ext4 (rw,relatime,seclabel,data=ordered)

# mount /p4      ※マウントポイントのみを指定してマウント

# mount
. . .
/dev/sdb5 on /p5 type ext4 (rw,relatime,seclabel,data=ordered)
/dev/sdb4 on /p4 type xfs (rw,relatime,seclabel,attr2,inode64,noquota)      ※マウントされた
```

- mountコマンドに「-a」オプションを指定すると、/etc/fstabファイルに記述された全てのファイルシステムをマウントします（noautoを除く）。

mount -a でマウント

```
# umount /p4 /p5      ※一旦、アンマウントする
# mount
. . .
tmpfs on /run/user/0 type tmpfs (rw,nosuid,nodev,relatime,seclabel,size=792148k,mode=700)

# mount -a          ※/etc/fstabファイルに記述された全てのファイルシステムをマウント

# mount
. . .
/dev/sdb4 on /p4 type xfs (rw,relatime,seclabel,attr2,inode64,noquota)  ※マウントされた
/dev/sdb5 on /p5 type ext4 (rw,relatime,seclabel,data=ordered)          ※マウントされた
```


- CD/DVDやUSBメモリ等のリムーバブルメディアを利用する時にも、それらをマウントすることが必要です。GNOMEデスクトップのようなGUI環境の場合には、メディアを挿入すると自動的にマウントされますが、CUI環境の場合には、mountコマンドでマウントします。

CD-ROMをマウントする

```
# mount -t iso9660 -o ro /dev/sr0 /media          ※読み取り専用でマウントする

# mount | grep iso9660
/dev/sr0 on /run/media/root/CentOS_7_x86_64 type iso9660 (ro,nosuid,nodev,relatime,uid=0,

# umount /media          ※取り出す前にアンマウントする

# mount | grep iso9660
```

- dfコマンドは、マウントされているファイルシステムの使用状況を表示します。

df コマンド

df [オプション]

オプション

説明

-h	M,G等の単位をつけて見やすく表示する
-i	iノードの使用状況を表示する

ファイルシステムの使用状況を表示する

```
# df -h
```

ファイルシス	サイズ	使用	残り	使用%	マウント位置
...					
/dev/mapper/centos-root	50G	5.5G	45G	11%	/
/dev/sdb2	1014M	172M	843M	17%	/boot
/dev/sdb1	200M	12M	189M	6%	/boot/efi
tmpfs	774M	36K	774M	1%	/run/user/0
/dev/sdb4	72G	33M	72G	1%	/p4
/dev/sdb5	59G	53M	56G	1%	/p5

ファイルシステムの iノード使用状況を表示する

```
# df -i
ファイルシス      Iノード      I使用      I残り      I使用%      マウント位置
. . .
/dev/mapper/centos-root    26214400    137923    26076477    1%      /
/dev/sdb2                  524288      61        524227      1%      /boot
/dev/sdb1                  0           0         0           -       /boot/efi
tmpfs                     990183      22        990161      1%      /run/user/0
/dev/sdb4                  37451776    4         37451772    1%      /p4
/dev/sdb5                  3932160     11        3932149     1%      /p5
```

- du コマンドはファイルやディレクトリのディスク使用量を表示します。

du コマンド

```
du [オプション] [ファイル | ディレクトリ]
```

オプション

説明

-h	M, G等の単位をつけて見やすく表示する
-s	総計だけをを表示する

ディレクトリのディスク使用量を表示する

```
# du /etc
. . .
0      /etc/scl/prefixes
0      /etc/scl
42620  /etc

# du -sh /etc
42M    /etc
```

■ free コマンドはメモリおよびスワップ領域の使用量を表示します。

free コマンド

free [オプション]

オプション

説明

-m	メガバイト単位で表示する (デフォルトはキロバイト単位)
----	------------------------------

メモリおよびスワップ領域の使用量を表示する

```
# free
```

	total	used	free	shared	buff/cache	available
Mem:	7921468	877712	5395228	207448	1648528	6570520
Swap:	8126460	0	8126460			

```
# free -m
```

	total	used	free	shared	buff/cache	available
Mem:	7735	858	5267	202	1609	6415
Swap:	7935	0	7935			

1. ディスク装置を利用するには、パーティション分割 → ファイルシステム作成 → マウントを実行します。
2. ファイルシステムには、XFS、ext4 他、多くの種類があります。
3. mkfsコマンドでファイルシステムを作成することができます。
4. スワップ領域を作成するには mkswapコマンド、有効化には swaponコマンド、無効化には swapoffコマンドを使用します。
5. mountコマンドは、マウント状況の表示と、ファイルシステムのマウントを行います。アンマウントするには umountコマンドを使用します。
6. システム起動時にファイルシステムが自動的にマウントされるようにするには、/etc/fstabファイルに設定を記述します。
7. リムーバブルメディアを利用する時にもマウントを行います。CUI環境では、mountコマンドでマウントします。
8. dfコマンドは、マウントされているファイルシステムの使用状況を表示します。

ご清聴ありがとうございました