

4章 デバイス/ファイルシステム/ファイルシステム階層標準

4-1. パーティションとファイルシステムの作成

4-1-1. パーティション管理方式

- パーティションを構成し、
- パーティションにファイルシステムを作成し、
- マウントして利用する。

- パーティションの管理方式
 - MBR
 - 管理コマンド : fdisk
 - 最大パーティションサイズに制約がある
 - パーティション数に制約あり
 - GPT
 - 管理コマンド : parted
 - MBRより制約が少ない
- MBRの管理
 - 基本パーティション
 - 一戸建て
 - 拡張パーティション
 - アパート
 - 論理パーティション
 - アパート内の部屋

4-1-2. ディスクのデバイスファイル

- /dev/sda1
 - s : 「h」 IDE形式のディスク、「s」 SCSI/USB/SATA形式のディスク
 - a : ディスクの台数。1台目をa、2台目をbというように指定
 - 1 : パーティション番号。1つめのパーティション、論理は5以降を指定。

4-1-3. fdiskコマンド

- fdisk
 - MBRパーティションを構成

```
fdisk [option] デバイスファイル
-1 : 指定したディスクのパーティション構成を表示
```

```
[root@localhost com]# fdisk -l /dev/sda
```

```
Disk /dev/sda: 8589 MB, 8589934592 bytes, 16777216 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O サイズ (最小 / 推奨): 512 バイト / 512 バイト
Disk label type: dos
ディスク識別子: 0x000bb087
```

デバイス	ブート	始点	終点	ブロック	Id	システム
/dev/sda1	*	2048	2099199	1048576	83	Linux
/dev/sda2		2099200	16777215	7339008	8e	Linux LVM

- fdiskコマンドの主なサブコマンド
 - p: パーティションの表示
 - n: パーティションの作成
 - d: パーティションの削除
 - w: 設定情報を書き込み終了
 - q: 設定情報を書き込まず終了
 - m: ヘルプの表示

4-1-4. partedコマンド

- parted (=gdisk)
 - MBRパーティションもしくはGPTパーティションを構成

```
parted [option] [デバイスファイル]
-1 : 指定したディスクのパーティション構成を表示
```

- partedコマンドの主なサブコマンド
 - mklabel: パーティションの管理方式をmsdos(MBR)、gptで指定
 - mkpart: パーティションの作成
 - rm: パーティションの削除
 - p: パーティションの表示
 - q: partedの終了。構成した内容の反映

4-1-5. ファイルシステムの作成

ファイルシステムの作成 = Windowsのフォーマット

- Linuxで利用できる主なファイルシステム
 - ext2: 従来標準として利用されていたファイルシステム
 - ext3: ext2 + ジャーナリング機能
 - ext4: ext3の最大ファイルシステムサイズの制約などを改良したもの(CentOS 6)
 - xfs: 高性能なジャーナリングファイルシステム。(CentOS 7)
 - iso9660: CDやDVDなどのメディア

- vfat : Windowsなどで利用されてきたファイルシステム

※「ジャーナリング」は、ジャーナルと呼ばれるデータを定期的に記録する技術のことで、ファイルシステムで、更新内容だけをジャーナルに記録する

- mkfs
 - ファイルシステムを作成

```
mkfs [option] デバイスファイル
-t ファイルシステム : 指定したファイルシステムを作成
```

- mke2fs
 - ext2/ext3ファイルシステムを作成

```
mke2fs [option] デバイスファイル
-j : ext3ファイルシステムを作成
```

4-1-6. スワップ領域の利用

データ領域のほか、**スワップ領域**にも独立したパーティションを割り当てられる
「mkswap」 -> 「swapon」で有効になる

- mkswap
 - スワップ領域を作成

```
mkswap [option] デバイスファイル
```

- swapon
 - スワップ領域の状態を確認、もしくは有効化

```
swapon [option] [デバイスファイル]
-s : スワップ領域の状態を表示
```

4-2. ファイルシステムのマウントとアンマウントの制御

4-2-1. マウントの実行

- マウントには「mount」コマンドを利用
- マウント先となるディレクトリをマウントポイントという

```
mount [option] [デバイスファイル] [マウントポイント]
-a : /etc/fstabでautoオプションが指定されているデバイスをすべてマウント
-t ファイルシステム : デバイスのファイルシステムを指定する
-o option : マウントオプションを指定する
    ro : 読み取り専用でマウントする
    remount : 一度アンマウントした後、再マウントする
```

- /proc/mounts や /etc/mtab は特殊なファイルでマウント情報が変更されるとファイルの内容も書き換わる。

4-2-2. /etc/fstabファイル

mountコマンドで実行したものは、システム起動中のみ有効。

次回以降もマウントしたい場合は、/etc/fstabという設定ファイルにマウント情報を記述する必要がある

- デバイス
 - マウントするデバイスを指定
- マウントポイント
 - マウントに利用するディレクトリを指定
- ファイルシステム
 - マウントする領域のファイルシステムの種類を指定
- オプション
 - マウントオプションを指定
- dump
 - 0だとdumpの対象外、1だとdumpの対象である
- fsck
 - 0だとfsckの対象外、1以降だと起動時のfsckの対象であると指定

◎主なマウントオプション

- defaults
 - デフォルトのオプション。オプションをまとめて指定するのと同じ意味 (async, auto, dev, exec, nouser, rw, suid)
- sync
 - ファイルシステムへの入出力を同期で実施
- async
 - ファイルシステムへの入出力を非同期で実施
- ro
 - 読み取り専用でマウント
- rw
 - 読み書き可能な状態でマウント
- auto
 - システム起動時、mount -aを実行したときに自動的にマウント
- noauto
 - システム起動時、mount -aを実行したときに自動的にマウントしない
- user

- 一般ユーザでもマウントできるようにする。アンマウントはrootユーザとマウントしたユーザのみ
- users
 - 一般ユーザでもマウントできるようにする。アンマウントも誰でもできる。
- nouser
 - rootユーザだけマウントできる

4-2-3. リムーバブルメディアのマウント

```
mount -t iso9660 /dev/sr0 /media/cdrom
```

- umount
 - アンマウントを実行

```
umount [option] [デバイスファイル] [マウントポイント]  
-a : マウントされているデバイスをすべてアンマウント  
-t ファイルシステム : 指定したファイルシステムのデバイスをアンマウント
```

- eject
 - リムーバブルメディアの取り出し

```
eject [option] [名前]
```

4-3. ファイルシステムの整合性の保守

4-3-1. 使用サイズの確認

- df
 - ファイルシステムの使用/空き領域サイズを表示

```
df [option] [デバイスファイル/マウントポイント]  
-h : わかりやすいサイズで表示  
-i : iノードについての情報を表示  
-t ファイルシステム : 指定したファイルシステムのデバイスを対象にする  
-T : ファイルシステムの種類も表示
```

- iノード : ファイルの属性情報やデータの格納先を格納する領域
 - ファイル種別 : 通常ファイル、ディレクトリ
 - 所有者情報 : 所有者、所有グループのID
 - リンクカウント

- アクセス権
 - タイムスタンプ
 - ファイルサイズ
 - データの格納先 : ブロック番号
- du
 - ファイルシステムの使用サイズを表示

```
du [option] [ディレクトリ]
-h : わかりやすいサイズで表示
-s : 合計サイズを表示
```

4-3-2. ファイルシステムの検査

- fsck
 - ファイルシステムのチェック

```
fsck [option] [デバイスファイル]
-y : 論理エラーがあった場合に確認なしで修復する
```

- e2fsck
 - ext2/ext3/ext4ファイルシステムのチェック

```
e2fsck [option] [デバイスファイル]
-y : 論理エラーがあった場合に確認なしで修復する
```

システム起動時にfsckを実行する！

- tune2fs
 - ext2/ext3/ext4ファイルシステムの設定を確認/変更

```
tune2fs [option] デバイスファイル
-l : 指定したファイルシステムの情報を表示
-c 回数 : 指定した回数以上マウントされたら、次回起動時にfsckを実行
-i 日数 : 指定した間隔で、次回起動時にfsckを実行
-j : ext2をext3に変換
-L ラベル名 : ラベル名の指定
```

- dumpe2fs
 - ext2/ext3/ext4ファイルシステムの状態を出力

```
dumpe2fs [option] デバイスファイル
```

- debugfs
 - ext2/ext3/ext4ファイルシステムを対話的メニューでデバッグ操作

debugfs [option] デバイスファイル

4-3-3. xfsファイルシステム

- xfs_info
 - xfsファイルシステムの情報を表示

xfs_info [option] デバイスファイル/ファイルシステム

- xfs_metadump
 - xfsファイルシステムのメタデータをバックアップ

xfs_metadump [option] デバイスファイル ファイル

4-4. ディスククォータの管理

- ディスククォータ
 - ファイルシステムごとに、ユーザやグループが利用するディスクサイズを制限することができる
 - 順番は以下
 - マウントオプションで有効化
 - クォータの有効化
 - 制限するサイズを指定
 - ディスククォータ関連のマウントオプション
 - usrquota : ユーザクォータの有効化 : ユーザごとに制限
 - grpquota : グループクォータの有効化 : グループごとに制限

クォータを有効にする

- quotacheck
 - ファイルシステム領域の使用量をチェックし、ファイルに出力
 - 以下のファイルが生成される
 - ユーザクォータ : aquota.user
 - グループクォータ : aquota.group

```
quotacheck [option] ファイルシステム
```

- quotaon
 - ディスククォータの有効化

```
quotaon [option] ファイルシステム  
-p : クォータの状態を表示
```

- edquota
 - ディスククォータの制限値を設定
 - ソフトリミット：警告が表示
 - ハードリミット：書き込みが禁止される
 - blocksはファイルサイズによる制限
 - inodesはファイル数による制限

```
edquota [option] [ユーザ/グループ]  
-t : 猶予期間を設定
```

4-4-2. ディスククォータの状態の確認

- quota
 - ユーザ\グループのクォータ制限の状態を表示

```
quota [option] ユーザ/グループ
```

- requota
 - ファイルシステムのクォータ制限の状態を表示

```
requota [option] ファイルシステム
```

- quotaoff
 - ファイルシステムのクォータ制限を無効にする

```
quotaoff [option] ファイルシステム
```

4-5. ファイルのパーミッションと所有者の管理

4-5-1. 所有者とパーミッション

- 読み取り : 4 : r
- 書き込み : 2 : w
- 実行 : 1 : x

4-5-2. パーミッションの変更

- パーミッションの変更
 - アルファベットや記号で指定する**シンボルモード**
 - 数値で指定する**オクタルモード**

```
chmod [option] mode ファイル/ディレクトリ
-R : 指定したディレクトリ以下のファイル/サブディレクトリもすべて変更する
```

対象	操作	権限
u : 所有者	+ : 追加	r : 読み取り
g : 所有グループ	- : 削除	w : 書き込み
o : その他のユーザ	= : 設定	x : 実行
a : すべての対象		s : SUID/SGID、t : スティッキービット

4-5-3. 所有者/所有グループの変更

```
# 所有者を変更
chown [option] 所有者[:所有グループ] ファイル/ディレクトリ
-R : 指定したディレクトリ以下のファイル/サブディレクトリもすべて変更する
```

```
# 所有グループを変更
chgrp [option] 所有グループ ファイル/ディレクトリ
-R : 指定したディレクトリ以下のファイル/サブディレクトリもすべて変更する
```

4-5-4. マスク値の設定

- パーMISSIONの設定。umaskコマンドを利用して、引き算のように設定できる

```
umask [option] [マスク値]
```

- * ディレクトリは規定では「777」
- * ファイルは規定では「666」
- * 664のファイルを生成するなら

```
umask 002
umask file
```

4-5-5. 特殊なパーミッション

- SUID
 - 所有者の権限で操作が可能
 - 4000
 - u+s
- SGID
 - 所有グループの権限で操作が可能
 - 2000
 - g+s
- スティッキービット
 - 所有者のみが削除可能
 - 1000
 - o+t

4-6. ハードリンクとシンボリックリンクの作成

- ln
 - リンクの作成

```
ln [option] リンク元 リンク
-f : リンク先が存在しても上書きして設定
-s : シンボリックリンクを作成（規定ではハードリンクを作成）
```

※ハードリンクを作ったのちコピー元のファイルを移動した場合、(つまりファイルシステム間を移動した場合、)コピー元のiノードが変わる。そして、ハードリンクの内容を書き換えるとハードリンクも新しいiノードが採番される。

```
[root@localhost link]# vi test02.txt
[root@localhost link]# ls -li
9552722 -rw-r--r--. 1 root root 5  4月  7 18:53 test02.txt
[root@localhost link]# ln test02.txt test02.link
[root@localhost link]# ls -li
9552722 -rw-r--r--. 2 root root 5  4月  7 18:53 test02.link
9552722 -rw-r--r--. 2 root root 5  4月  7 18:53 test02.txt
[root@localhost link]# mv test02.txt /boot
[root@localhost link]# ls -li
9552722 -rw-r--r--. 1 root root 5  4月  7 18:53 test02.link
[root@localhost link]# vi test02.link
[root@localhost link]# ls -li
9552723 -rw-r--r--. 1 root root 12  4月  7 18:54 test02.link
```

※シンボリックリンクは、iノードではなくファイル名に紐づくので、ファイル名を変えると機能しなくなる

※シンボリックリンクは、ディレクトリにも適応できる

※ハードリンクは別のファイルシステムには作れない（シンボリックリンクは作れる）

4-7. システムファイルの確認と適切な位置へのファイルの配置

4-7-1. FHS

- FHS(Filesystem Hierarchy Standard : ファイルシステム階層標準)
- ディレクトリ構成の基準のこと

代表的なファイルシステムに格納されているもの

- /dev : ハードディスク、DVD-ROMなどのデバイスファイル
- /etc : システム、コマンドなどの設定ファイル
- /lib : 共有ライブラリやモジュール、/bin、/sbinにあるコマンドが利用するライブラリ
- /opt : 追加パッケージや追加プログラム
- /root : rootユーザのホームディレクトリ
- /bin : 一般ユーザが実行できるシステムコマンド(ls, catなど)
- /sbin : rootユーザの使用するシステムコマンド(rebootなど)
- /usr : ユーザが共有するコマンド
- /usr/bin : 一般ユーザとrootユーザが使用する基本コマンド
- /usr/sbin : rootユーザが使用する基本コマンド
- /usr/lib : プログラムの共有ライブラリ
- /usr/local : 個人で作成したコマンド
- /usr/src : Linuxのカーネルソースなどのソースコード
- /media : DVD-ROMなどのリムーバブルメディアのマウントポイント
- /mnt : 一時的にマウントするファイルシステムのマウントポイント
- /proc : カーネル内部の情報にアクセスする
- /tmp : 一時ファイルが配置され、すべてのユーザーが読み書きできる
- /home : 各ユーザーが利用する専用のホームディレクトリ
- /boot : 起動に必要なカーネルイメージ
- /var : ログファイルなどの頻繁に書き込まれるファイル
- /var/cache : 一時的なキャッシュファイル
- /var/lock : アプリケーションを制御するためのロックファイル
- /var/log : ログファイル
- /var/run : システム状態を示すファイル
- /var/spool : 印刷待ちのデータ、予約されたジョブ

4-7-2. ファイルの検索

- find
 - -name
 - -atime (最終アクセス日 -2 -3 +1 など)
 - find ./ -atime -2 -type f
 - -u
 - -l

- -mtime (最終更新日時)
- -perm パーミッション
- -size
- -user
- -type (fならファイル)
- -maxdepth 数 (階層を指定)
- -mindepth 数
- -exec コマンド {} ;
 - find ./ -mtime +1 -type f -exec rm {} ;

find [検索ディレクトリ] [option] [条件式] [アクション]

現在から3日前まで (現在時間～72時間前)

```
# find ./ -mtime -3
```

3日前 (72時間前～96時間前)

```
# find ./ -mtime 3
```

過去から3日前まで (72時間前～過去)

```
# find ./ -mtime +2
```

- xargs
 - 標準入力に送られてきた対象に対して、コマンドを実行

```
find /tmp -type f | xargs echo
```

- locate (findより高速に検索が可能 : updatedbコマンドを更新する必要がある)
 - /etc/updatedb.conf
 - 検索対象から外す
 - PRUNE_BIND_MOUNT
 - yes: mount --bindでマウントした領域を対象外
 - PRUNEFS
 - 指定したファイルシステムを対象外
 - PRUNENAMES
 - 指定した文字列が含まれるファイルを対象外
 - PRUNEPATHS
 - 指定した文字列が含まれるパスを対象外

4-7-3. コマンドやマニュアルパスの格納先

- which
- type
- whereis