

システムプログラム 第7回

創域理工学部 情報計算科学科

松澤 智史

本日の内容

- システムプログラムの一部であるOSについて学ぶ
 - 第7回では, OSの機能の一つである「ファイルシステム管理」を学ぶ

計算機に保存されるデータ

0,1のビット列が直列に保存される



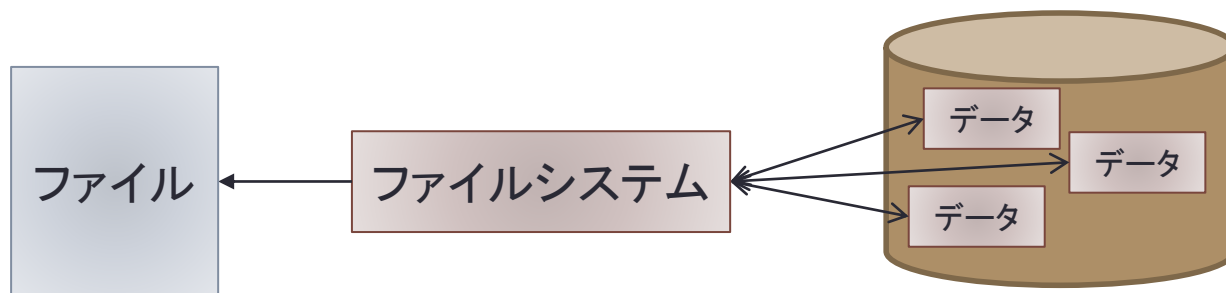
01011110111000011.....

どのデータがどこに格納される？
データの追加・削除はどのように管理？

ファイルという抽象的な概念を用いて管理する
というお話し

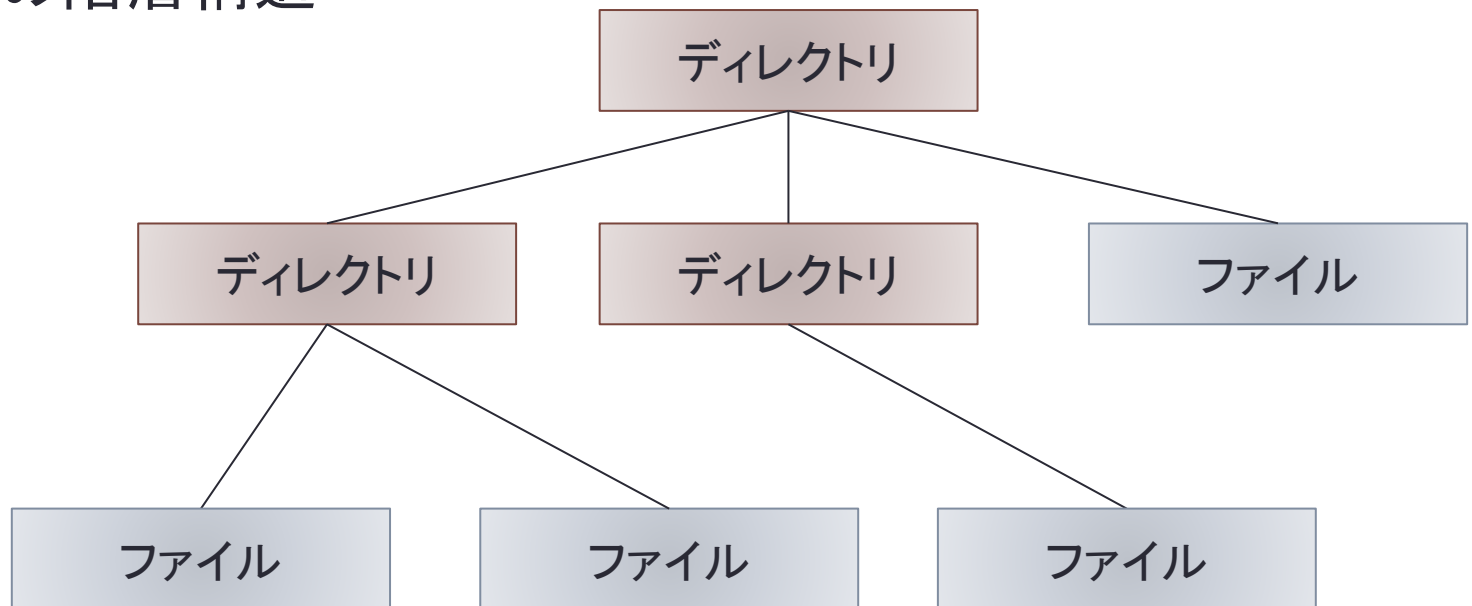
ファイルシステムの概要

- ファイルというインタフェースをユーザやアプリケーションに提供するシステム
- 記憶装置にデータがどのように記録されているのかを意識することなくデータをファイルという仮想的な容器に格納・取り出しが可能
- ファイルの入れ物である「ディレクトリ」を階層的に配置
- ファイルの管理性の向上
- 記録データのセキュリティの確保



ディレクトリ

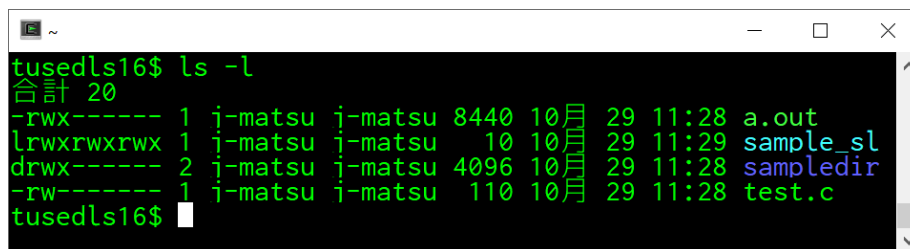
- ファイルの一種
- ファイルやディレクトリを格納する容器
- ツリー上の階層構造



ファイルのメタ情報

- 実データとは別にファイルにはメタ情報(メタデータ)が存在する
- UNIX系ファイルシステムではi-nodeと呼ばれるメタ情報がある

代表的なi-node情報

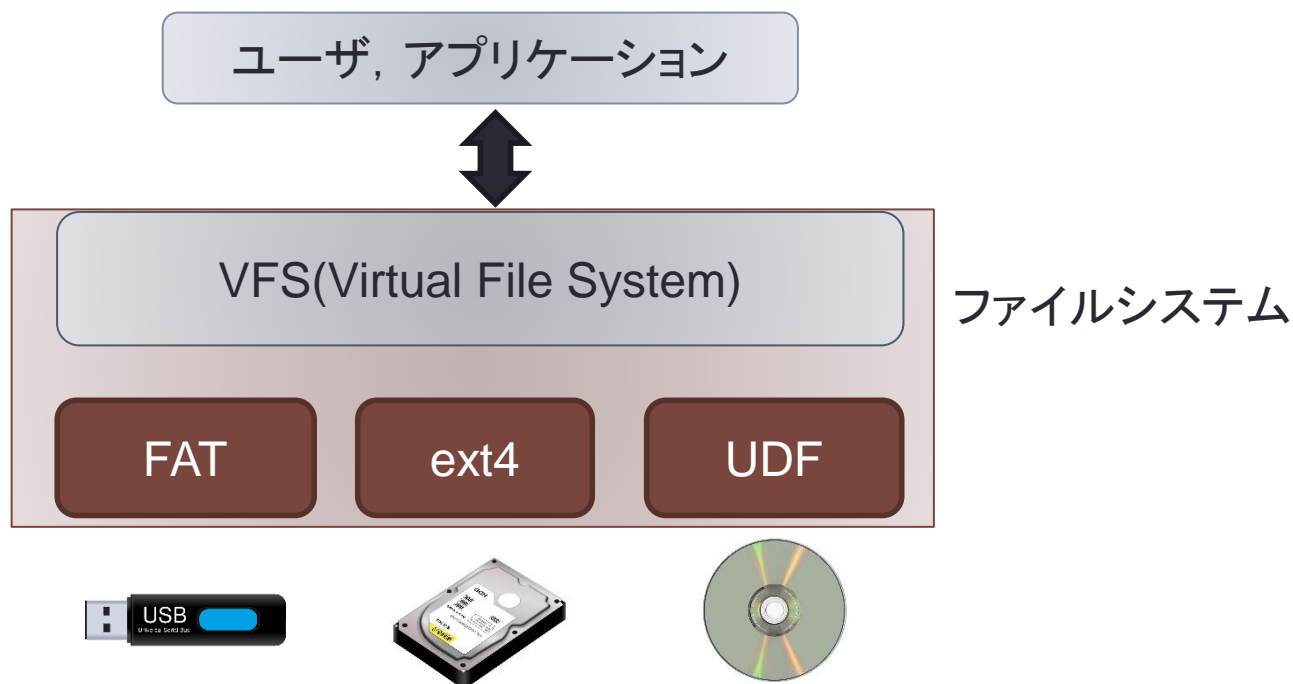


```
tusedls16$ ls -l
合計 20
-rwx----- 1 j-matsu j-matsu 8440 10月 29 11:28 a.out
lrwxrwxrwx 1 j-matsu j-matsu  10 10月 29 11:29 sample_sl
drwx----- 2 j-matsu j-matsu 4096 10月 29 11:28 sampledirt
-rw----- 1 j-matsu j-matsu  110 10月 29 11:28 test.c
tusedls16$
```

i-node情報	意味
ファイルタイプ	通常ファイル, ディレクトリなどを識別するためのフラグ
ファイルモード	ファイルアクセス権(読み, 書き, 実行)
所有者番号	ファイル所有者ユーザーのID
グループ番号	ファイル所有者ユーザーのグループID
ファイルサイズ(byte)	ファイルのサイズ
ファイル参照時刻	ファイルのデータを参照した時刻
ファイル更新時刻	ファイルのデータを更新した時刻

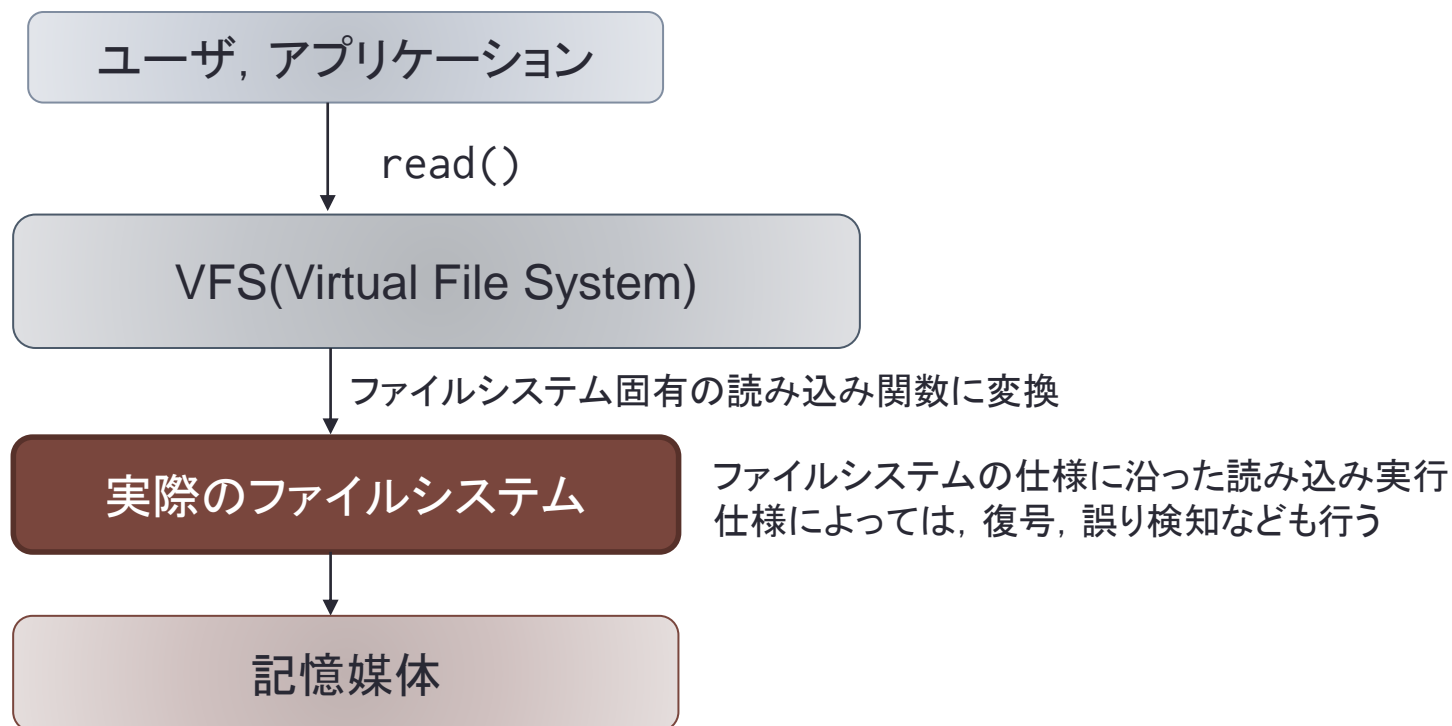
VFS(Virtual File System)

- ファイルシステムには実際の記憶装置(ディスク)に依存した様々な種類がある
- VFSは様々な種類のファイルシステムの違いを吸収して統一したインタフェースを提供する
 - ファイルオープンやアイコンダブルクリックなどファイルの読み、書き、実行の手順はファイルシステムに依存しない



VFS(Virtual File System)

- ユーザやアプリケーションにwrite(), read()などの関数を提供している
- write(), read()などの関数が呼ばれた場合は、
実際のファイルシステムの読み書きの関数を必要に応じて使用する



Linuxカーネルがサポートするファイルシステム

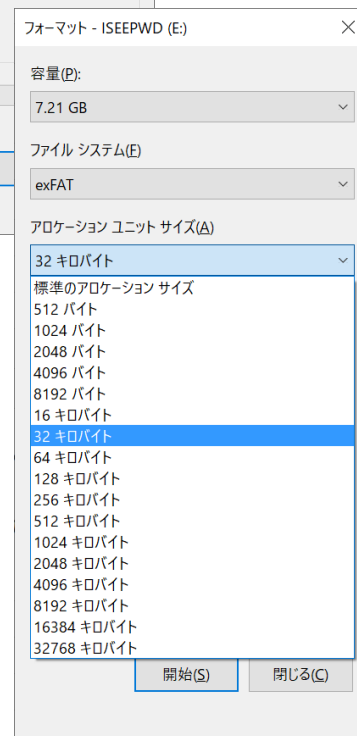
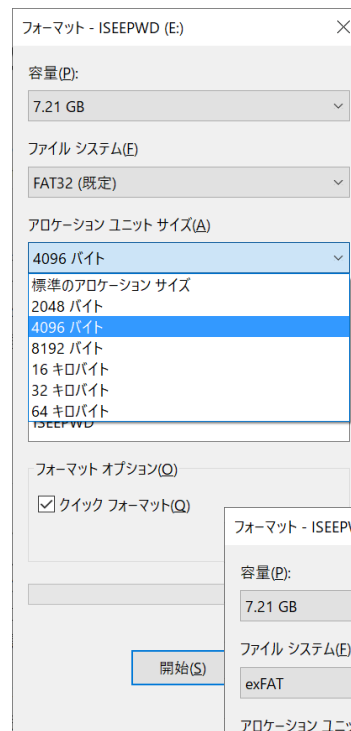
ファイルシステム	概要
Btrfs	RAID機能
CIFS	Windows ファイル共有
Ext2,3,4	Linux標準
FAT, exFAT	SDカード, USBメモリ, 昔のWindows
HFS+	MacOS
ISO9660	CD-ROM
NFS	ネットワークファイルシステム
NTFS	Windows標準
tmpfs	メモリ上に配置
UDF	CD/DVD
XFS	大容量ストレージ向け

FAT(File Allocation Table)

- 用語
 - セクタ HDDが一度に読み書きするサイズ(通常512バイト)
 - クラスタ 複数のセクタをまとめたもの
- FAT12, FAT16, FAT32
 - クラスタを管理するFATエントリのビット幅によって3種類
 - クラスタ・データ領域を一定のサイズ(512バイト~64kバイト)に区切った単位
 - FAT12 クラスタ数が4085以下 ~128M
 - FAT16 クラスタ数4086-65525以下 2M~2G
 - FAT32 クラスタ数65526以上 32M~2T
- クラスタのサイズと数によって性能が決まる
 - クラスタごとに読み書きする(1バイトのデータでもクラスタサイズの領域を使う)
 - クラスタサイズを小さくしたほうが
 - ディスクの隙間(クラスタギャップ)が少なくて済む
 - FATへのアクセスが頻繁に発生し全体のアクセス速度が遅くなる

FAT, exFAT

- WindowsでUSBメモリやSDカードのフォーマット
- アロケーションユニットサイズ = クラスタサイズ
 - 複数のクラスタにデータがまたがる場合は、次クラスタの位置を保持している(クラスタサイズごとに位置情報の読み込みが必要)
- 小さめのファイルをたくさん格納したければクラスタサイズを小さめにするが無駄な領域が減る
- クラスタサイズは大きくする方が読み込み速度が増す(大きめのファイルを少数格納する場合は無駄な領域はそもそも多くない)
- exFATにするとクラスタサイズを大幅に増やせる
 - FAT32は~2TBまで(Windowsは32GB制限あり)の容量であるがexFATの最大容量は256TBまで(理論上は16EB)

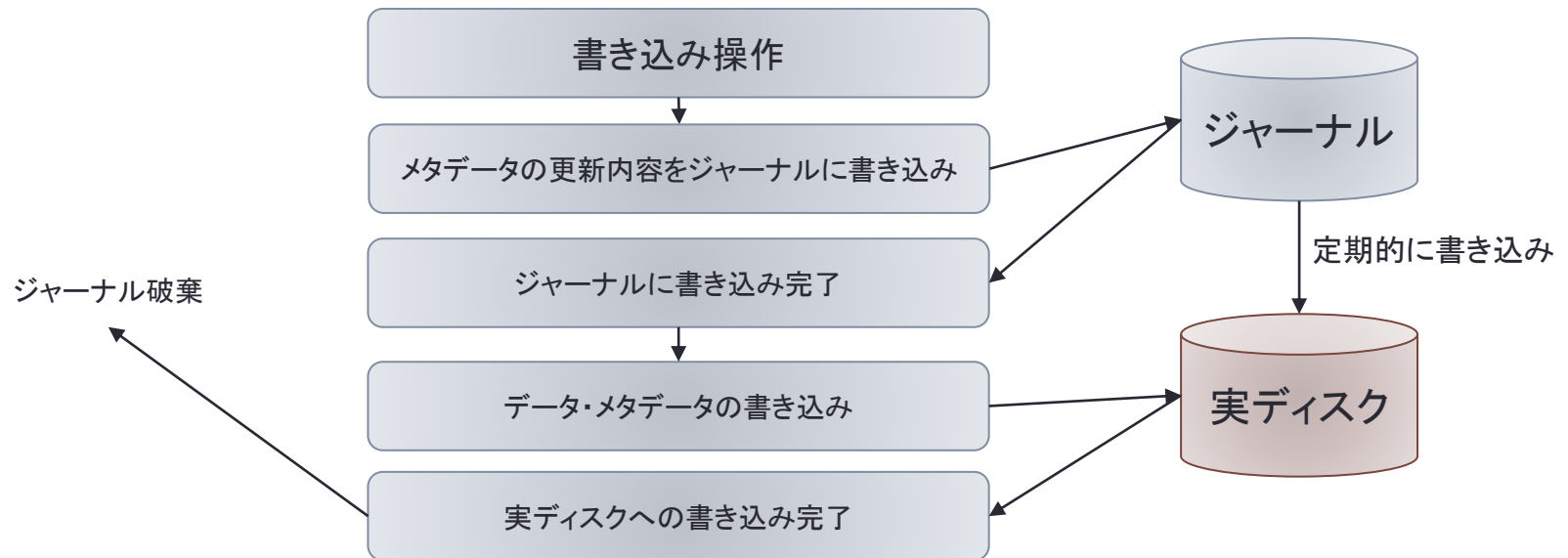


NTFS(Windows標準ファイルシステム)

- exFATと同じ最大容量(クラスタサイズ, クラスタ数)
 - 理論上4EB, 実際は256TB
 - クラスタサイズ4k(デフォルト)で16TB
- 1ファイルの最大サイズ制限なし(FAT32は1ファイルの最大サイズが4GB)
- ファイルやディレクトリに対して暗号化が可能
- ファイル単位で圧縮が可能(クラスタサイズ4k以下のみ)
- ジャーナリングファイルシステム

ジャーナリングファイルシステム

- データ保護の手法
- ジャーナルと呼ばれるデータを定期的に記録する
 - 更新内容だけをジャーナルに記録する
- 電源断によるシステムトラブル時にも復旧可能(部分的)

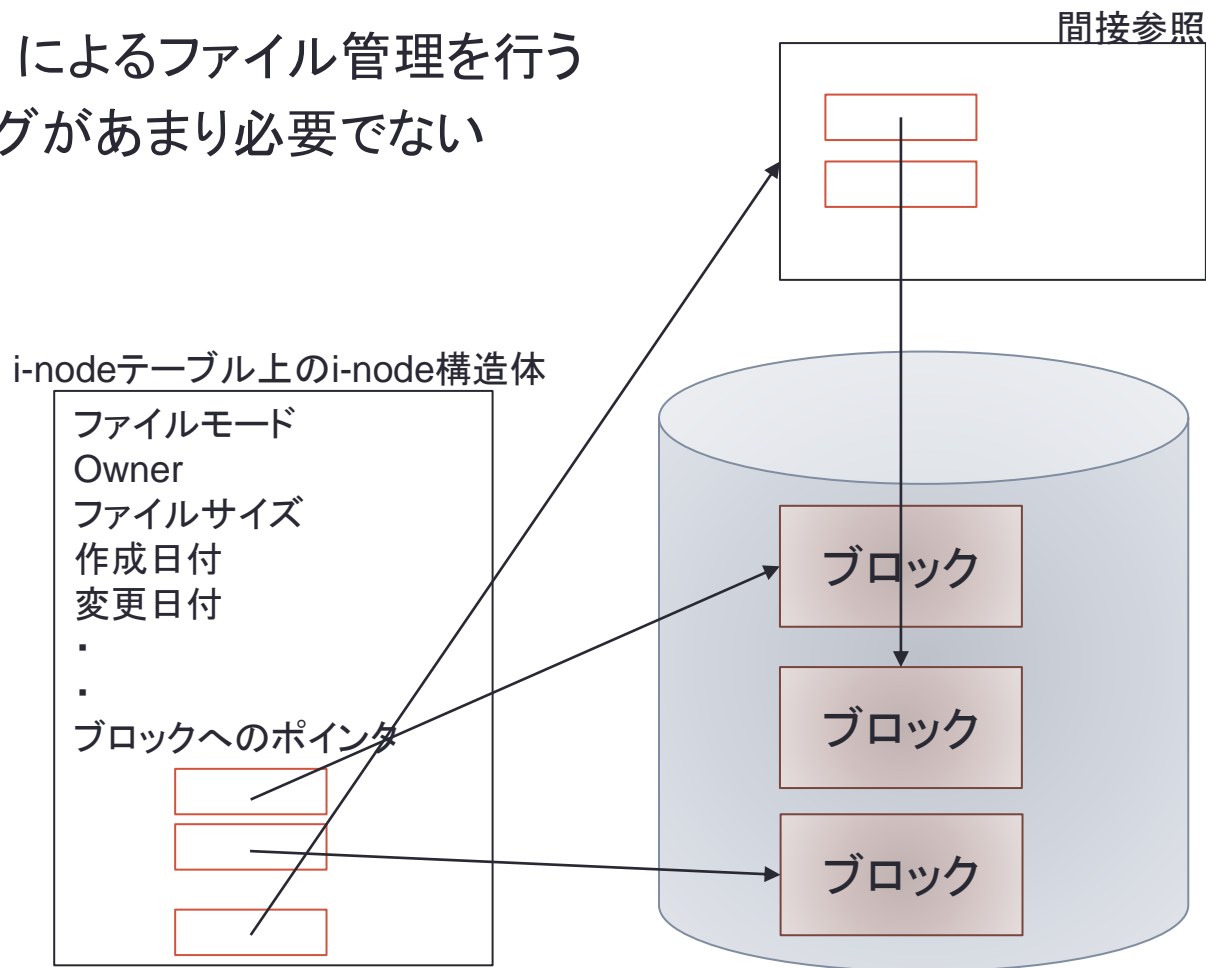


ジャーナリングファイルシステム

- トラブル時の処理
 - ジャーナルファイルを参照し、整合性の取れないファイルを破棄、または復旧する
- トランザクション処理
 - 2つ以上の処理は、両方実行されるかどちらも実行されないかの二者択一となる
 - 詳細はデータベースシステムの講義を参考のこと
- 一般的にジャーナルによって保護されるのはメタデータまで
 - ファイルの中身までは保護されない
 - NTFSも該当する

ext2, ext3, ext4 (Linux標準ファイルシステム)

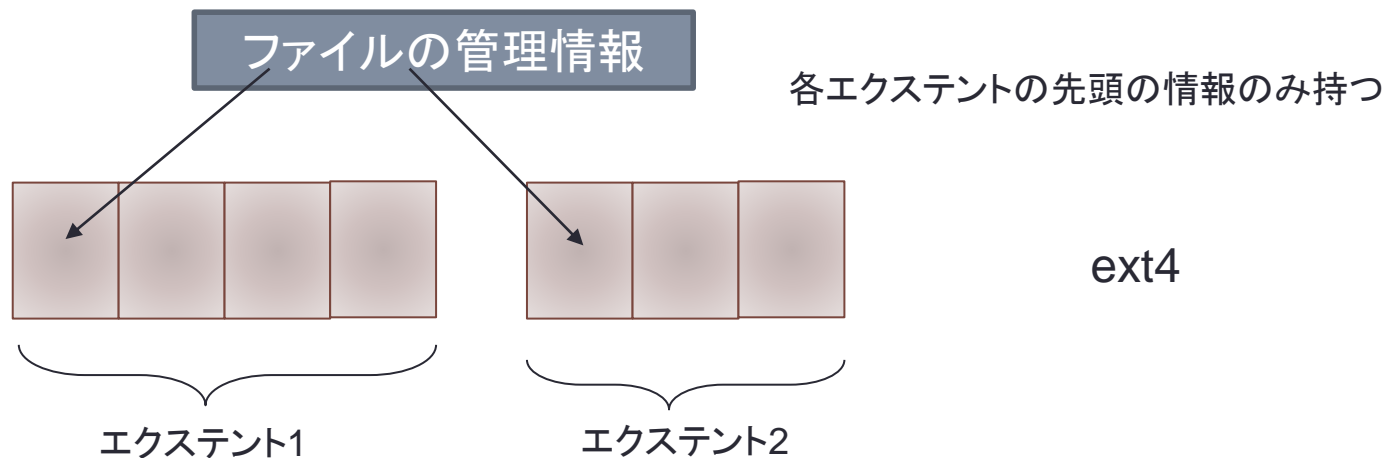
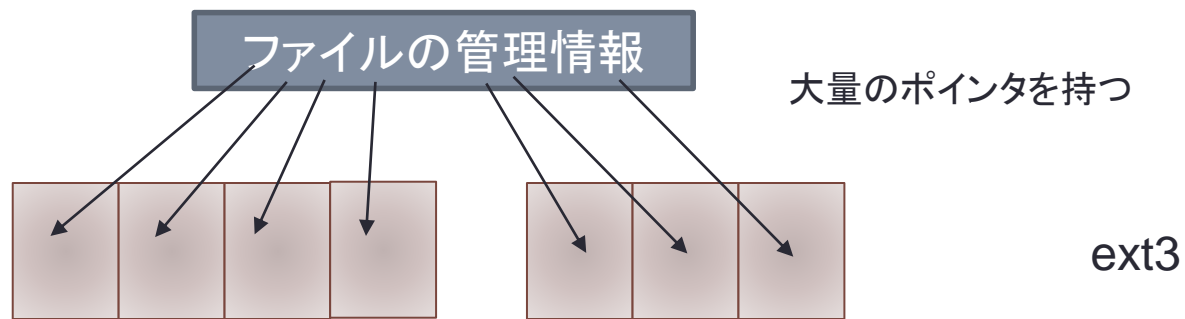
- i-node によるファイル管理を行う
- デフラグがあまり必要でない



ext2, ext3, ext4 (Linux標準ファイルシステム)

- ext2
 - 間接参照によって原理上1ファイル4TBまで(実際は2TB)
 - ジャーナリングファイルシステムは未実装
 - ディレクトリ構造は連結リストで管理
- ext3
 - ext2にジャーナリングを追加したファイルシステム
 - NTFSと異なり, ジャーナルだけでなくデータも保護対象
- ext4
 - 1ファイル16TB, 最大容量1EBまで対応
 - ディレクトリ構造をB-Treeの改良版であるH-Treeで管理
 - ジャーナルにチェックサムを導入
 - 間接参照の多段のオーバーヘッドを軽減するために隣接するブロックに対してその格納場所を配列として保持

ext4 エクステンツ



XFS(eXtents File System)

- LinuxのRedHat系が標準採用
- i-nodeによるファイル管理
- ジャーナリングファイルシステム
- 記憶領域を同一サイズの複数のエリアに分け、各アロケーショングループを独立に管理する
 - 書き込み処理を並列に行うことができる
 - 削除されたファイルの復元は不可能

HFS(Hierarchical File System)

- MacOSで使用されているファイルシステム
 - 拡張されたHFS+(MacOS拡張)もある
 - MacOS 10.13 以降はHFS+をさらに拡張したAPFS(Apple File System)
- 大文字と小文字の区別なし
 - Desktopとdesktopは同じファイル名
- 実際のデータ(データフォーク)とリソースフォークと呼ばれるメタ情報を保持
 - リソースフォークにはアイコンやウインドウの形状などの様々な情報が含まれる
- HFS+
 - ブロックサイズ4kとなり大容量ファイルを扱える
 - ファイル名がUnicode
 - フォークシステムの拡張
- APFS
 - 大文字と小文字の区別, 暗号化をサポート, などの拡張がされている

UDF(Universal Disk Format)

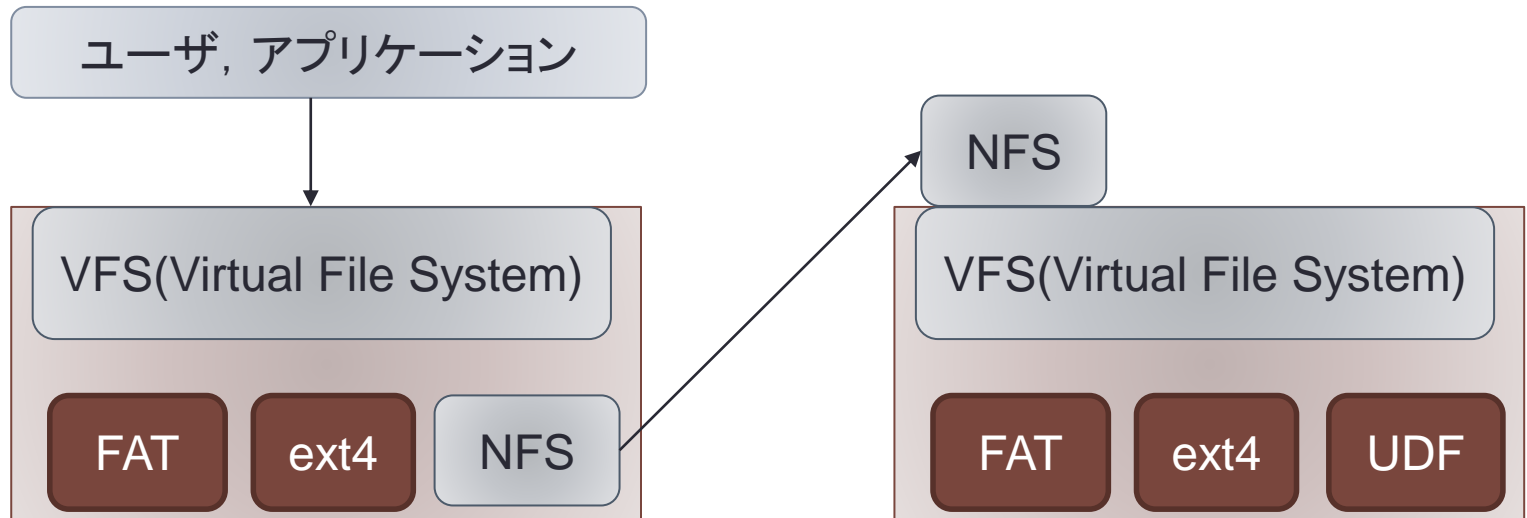
- CD/DVDのような光ディスクを対象としたファイルシステム
- 記録された情報の一部が破損しても、破損していない部分を読み出せる
 - 誤り訂正符号を用いている
 - リードソロモン符号
- 512バイト単位での書き込みと読み込み

tmpfs

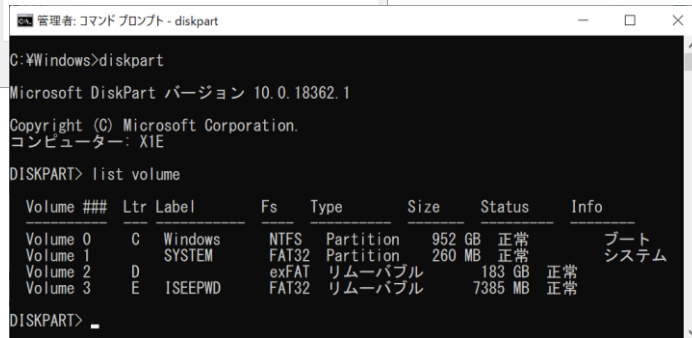
- 一時ファイルのための仕組み
- 他のUNIX系ファイルシステムと同様にi-nodeで管理する
- ディスクにデータを格納するかのように見せかけるが実際にはメモリに格納する
- リブートするとすべてのファイルは失われる

NFS(Network File System)

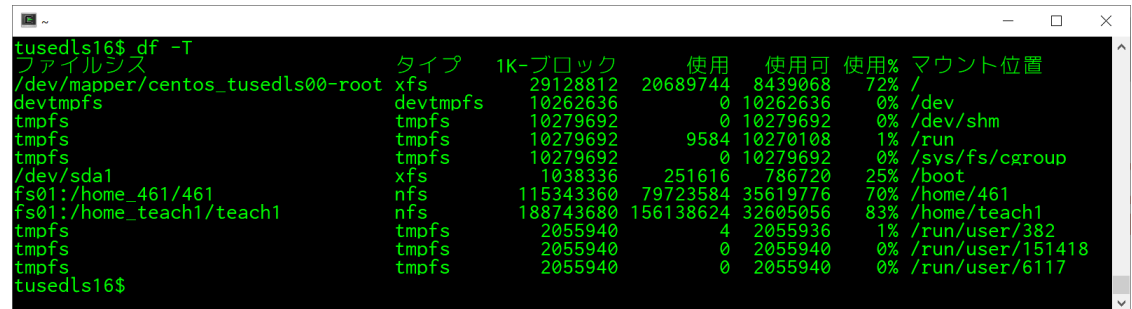
- ネットワーク共有型ファイルシステム
- 仮想のファイルシステム



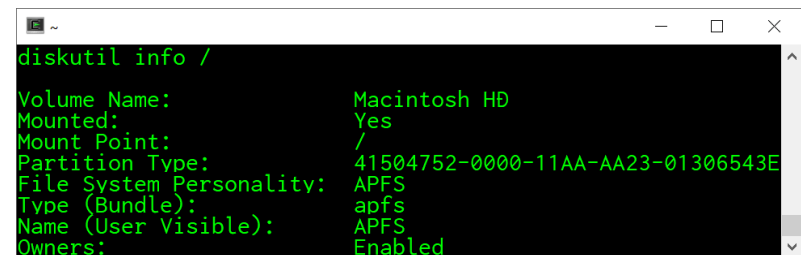
ファイルシステムの確認



Windows



Linux



Mac

どのファイルシステムが良い？

- ジャーナリングファイルシステムはリムーバルデバイス向きではない
 - NTFS ext4 HPS+ APFSなどは固定のSSD/HDD向き
 - ext4はメタデータのみならず実データ保護も行うジャーナリングファイルシステム
- FAT, exFATは書き込み中でなければいきなり切断しても原理上問題ない
 - クラスタサイズにもよるが、書き込みはexFATよりFAT32の方が速いことも多い
 - 大量のデータを扱うリムーバルデバイスはexFAT一択
- どのファイルシステムでもクラスタ(ブロック)サイズは一考すべき
 - 様々なサイズのファイルを扱うメインドライブは、大きくしすぎても小さすぎても非効率

ファイルシステムやクラスタサイズを変更して
ディスクへの読み書きの速度を比較してみよう

まとめ

- ユーザやアプリケーションからファイルという形でデータにアクセスさせるためのファイルシステムを学んだ
- ファイルシステムには、対象となるハードウェアを考慮した独自の仕様がある
- VFS(Virtual File System)が導入されていることによって実際に使用されているファイルシステムの差異を吸収する

質問あればどうぞ