



オペレーティングシステム

#14 ファイル:

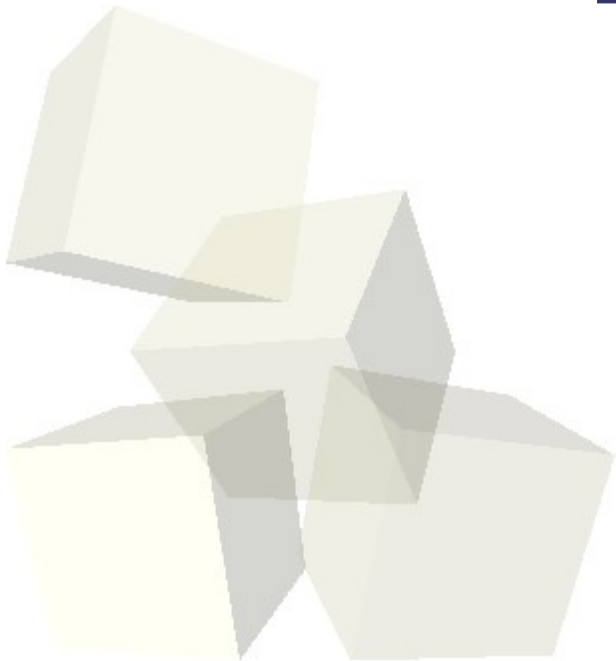
より進んだファイルシステム





14.1

ディスクキャッシュ



■ 仮想記憶

- 主記憶の大きさの制限をなくす
- そのために主記憶上のデータをディスク(二次記憶)に退避する

■ ディスクキャッシュ

- ディスクアクセスの遅さを改善する
- そのために頻繁に読み書きされるディスク上のデータ(ファイル等)を主記憶に置く

■ 書き込み発生時における一貫性保持

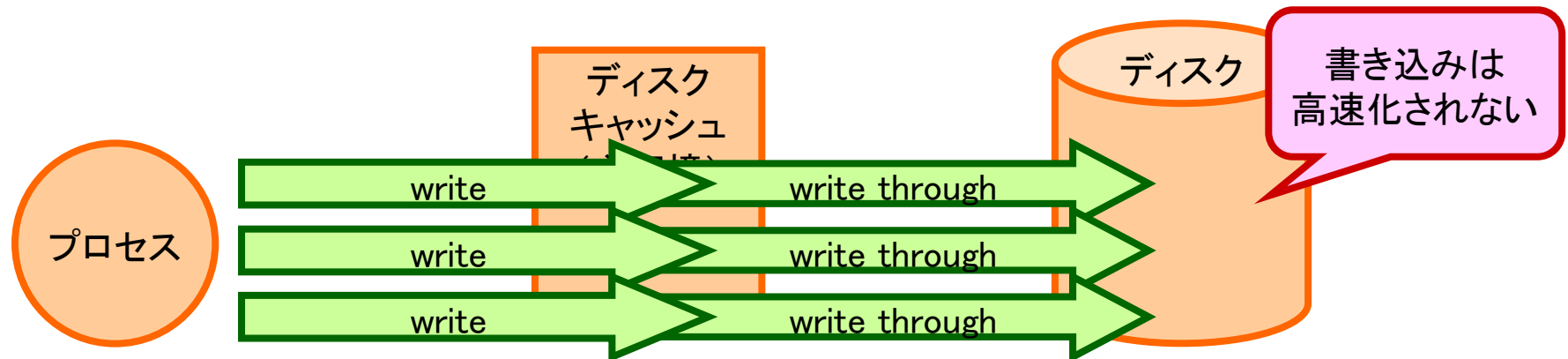
- 書き込みは主記憶上に存在する
ディスクキャッシュに対して行う
- 主記憶上データとディスク上データが異なってしまう
- 何らかの形で一貫性をとる必要がある

■ ライトスルー

■ ライトバック

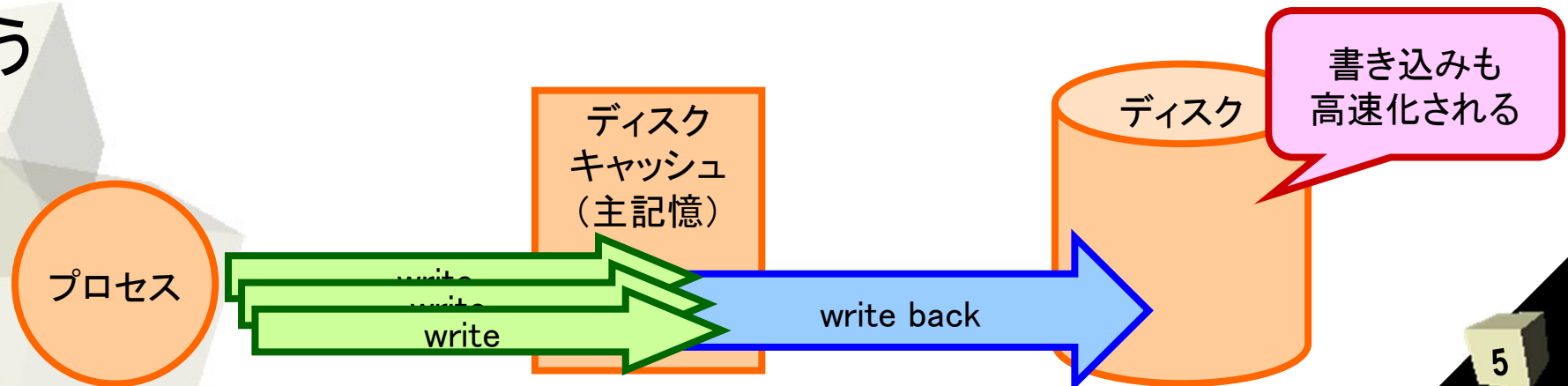
■ ライトスルー (Write Through)

- 書き込み発生時, ディスクに対して書き込みを行う



■ ライトバック (Write Back)

- 書き込み発生時はディスクキャッシュに対してのみ行う



- ファイルインデックス情報(FATなど)は、高速化のために、二次記憶から主記憶にコピーされ、更新は通常主記憶のみ行われる
 - 高速化のため
- ファイルインデックス情報は、
 - 二次記憶のマウント時に主記憶にコピー
 - 二次記憶のアンマウント時に書き戻す
- もし、インデックス情報を書き戻さなかった(書き戻せなかった)場合は？
 - 停電やOSの以上終了 etc

- 二次記憶の一部(固定領域)をジャーナル領域とする。
- 書き込み時
 - ジャーナル領域に、下記の情報を書き込む。
 - 空きセクタ確保情報
 - inode確保・更新情報
 - セクタ書き込み情報
 - ディレクトリ更新情報
 - 書き込み終了後、プロセスに書き込み完了通知(commit)
 - 非同期に、ジャーナル領域にあるファイル変更情報を用いて、ファイル実体を更新
- commit前に異常終了した場合、ファイルの書き込みがなかったものとして扱う
 - ファイル実体の変更はないので、不整合は発生しない。
- 非同期書き込みの際に停止した場合は
 - 再実行すればいい

■ Orderedモード

- データが、ファイル実体に反映された後で、そのデータを管理する**inode情報**、**ディレクトリ情報**のみをジャーナル領域に書き込む(LINUX Ext3ファイルシステムのデフォルト)

■ Writebackモード

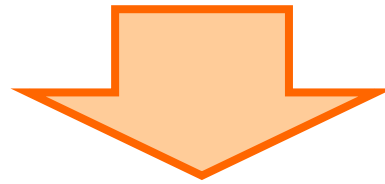
- データの保存領域への書き込み終了を待たずに、メタデータをジャーナル領域に書き込む
- 書き込み性能は向上するが、データの保存が失敗した場合でもジャーナル領域が更新される可能性

■ Journalモード

- データの書き込み情報もジャーナル領域に書き込む
- 確実だが、遅い

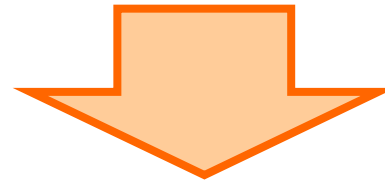
ジャーナルファイル時の読み書き

orderedでは書き込み終了は、メタデータをジャーナル領域にのみ書き込めば、通知可能（メタデータのファイル実体への書き込みは非同期）



書き込み速度 $\text{journal} < \text{ordered} < \text{writeback}$

journal設定では、読み込み時、ファイル実体に変更が反映されていないことも考慮する必要がある（ジャーナル領域の内容も確認する必要）。



メタデータ(ディレクトリなど)を操作するファイルの生成・削除時や、ファイル実体がcommitされていないファイルを読むときには、速度が少し遅くなる
読み込み速度 $\text{journal} < \text{ordered} \div \text{writeback}$

The screenshot shows the Windows Disk Management console and the properties window for a Canon MG7700 series USB device.

ディスクの管理 (Disk Management)

ボリューム	レイアウト	種類	ファイル システム	状態	容量	空き領域	空き領域の割...
(C:)	シンプル	ベーシック	NTFS	正常 (ブート...	1787.30 GB	1360.90 ...	76 %
(ディスク 3 パーティシ...	シンプル	ベーシック		正常 (EFI ...	99 MB	99 MB	100 %
UBUNTU 20_1 (I:)	シンプル	ベーシック	FAT				
ボリューム (E:)	シンプル	ベーシック	NTFS				
ボリューム (F:)	シンプル	ベーシック	NTFS				
ボリューム (G:)	シンプル	ベーシック	NTFS				
ボリューム (H:)	シンプル	ベーシック	NTFS				

ディスク 5 リムーバブル (L:)
メディアなし

ディスク 7 リムーバブル 3.73 GB オンライン
UBUNTU 20_1 3.73 GB FAT 正常 (アクティブ)

Canon MG7700 series USB Deviceのプロパティ

全般 ポリシー ボリューム ドライバー 詳細 イベント

取り外しポリシー

- ☒ クイック取り外し (既定)(Q)
デバイス上と Windows での書き込みキャッシュを無効にします。ただし、[安全な取り外し] 通知アイコンを使わなくてもデバイスを安全に切断できます。
- ☐ 高パフォーマンス(B)
Windows で書き込みキャッシュを有効にします。ただし、デバイスを安全に切断するには、[\[ハードウェアの安全な取り外し\]](#) 通知アイコンをクリックする必要があります。

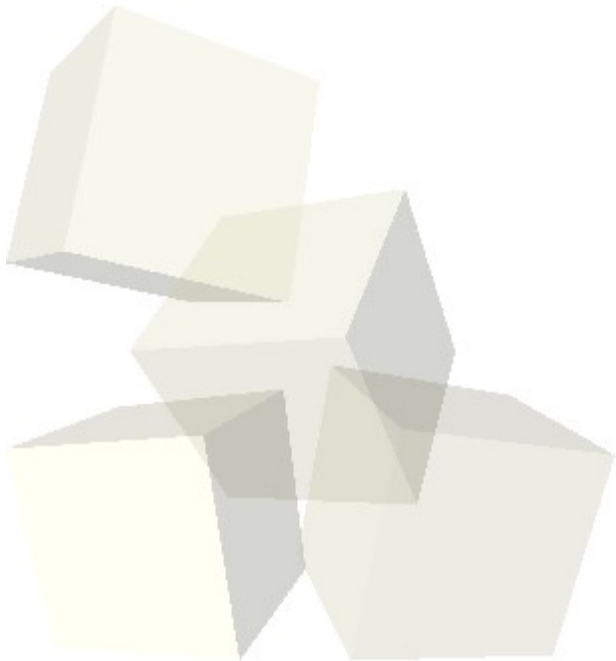
全ディスク ドライブ(D):

名前
Canon MG7700 serie
CSSD-M2B1TPG3VN
CyberLink Virtual Cd
Force MP510
GH PicoN USB Devi
HL-DT-ST BD-RE BH
Microsoft 仮想ディス
Samsung SSD 750 EV
WDC WD40EZRZ-00GXC
WDC WD40EZRZ-00GXC

OK キャンセル



14.2 非同期入出力



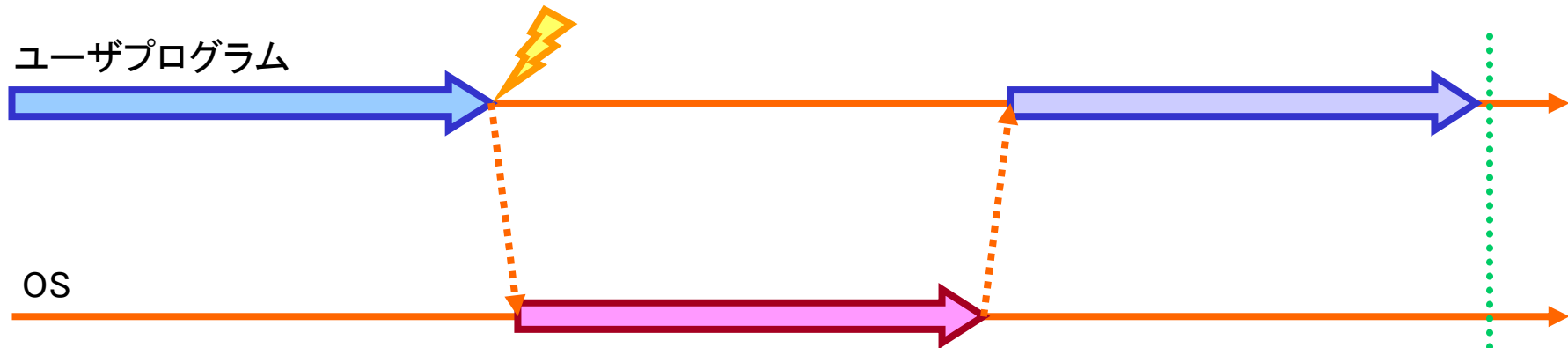
■ 通常の I/O (同期)

- スーパバイザコールにより
実行中のプロセスを待ち状態に移行
- I/O 操作
 - プロセスにとっては非常に長い待ち時間

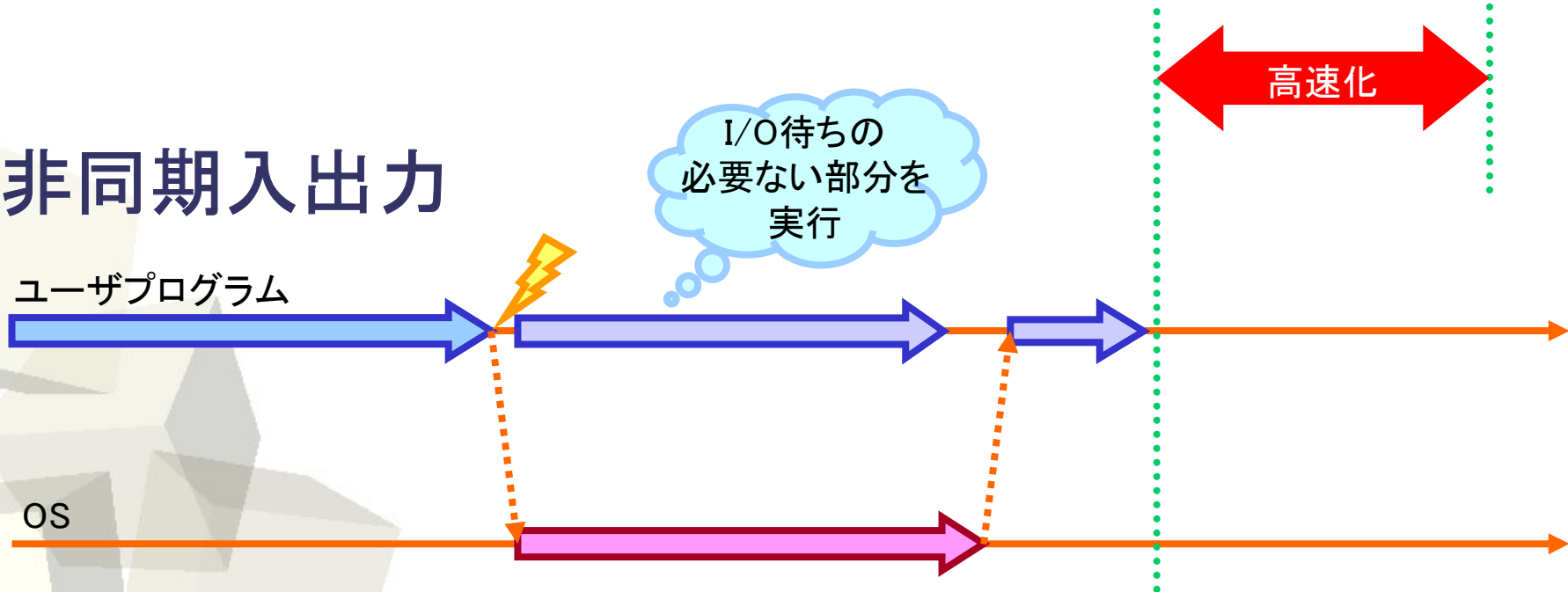
■ ノンブロッキング I/O (非同期)

- I/O 操作の際, プロセスを止めない
- I/O 処理を待たずに実行できる部分を先にやっつく
- 但し、I/O 処理が終わったかを確認するのは呼び出し側プロセスの責任

■ 同期入出力



■ 非同期入出力





14.3

ファイルシステムの仮想化



■ ディスクキャッシュ

- ディスク上のデータを一部主記憶に置くことで、低速なディスクへの読み書きを、主記憶アクセス時間で実現する

■ VFS

- アプリケーションと二次記憶装置との間に位置する
- アプリケーションからのファイル操作要求を、各二次記憶システムごとのAPI(Application Program Interface)に変換

■ ステージング

- テープ等のデータを一部ハードディスクに置くことで、低速な二次記憶への読み書きを、(比較的高速な二次記憶である)ハードディスクへのアクセス時間で実現する

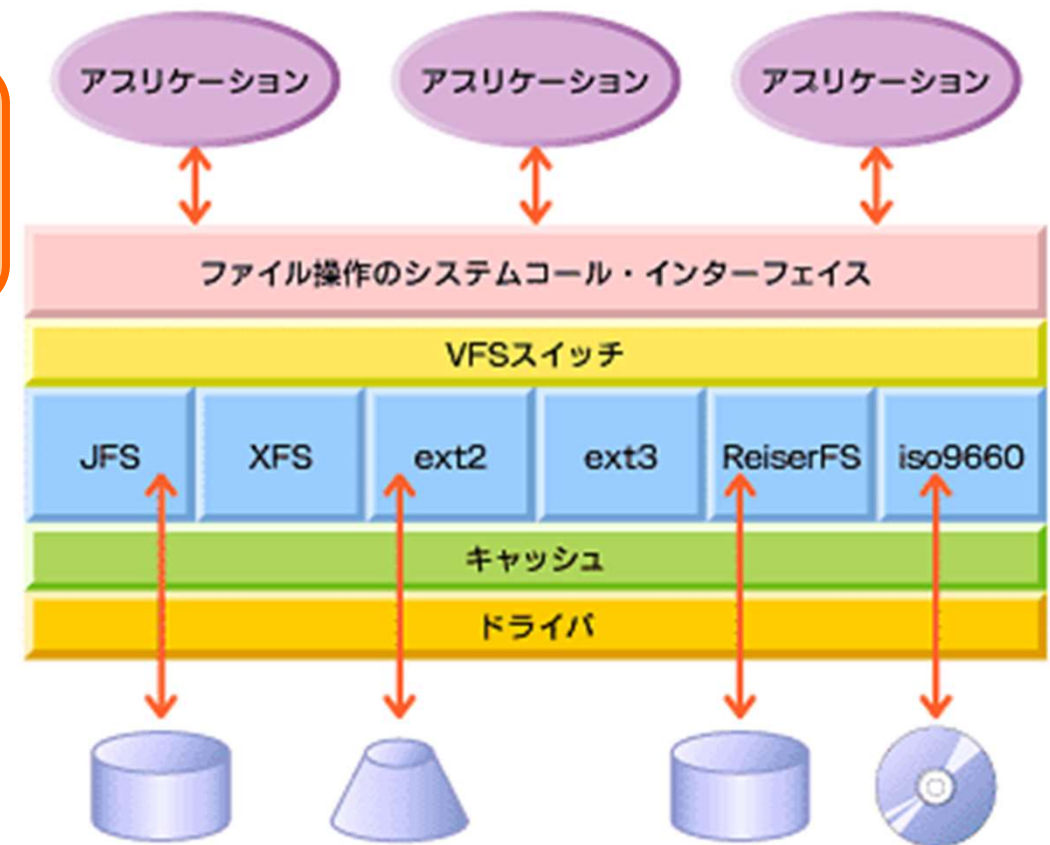


■ さまざまな二次記憶

- 速度や書き込み方法の違い
 - ハードディスク, SSD, CD/DVD
 - NFS, SMB (ネットワーク経由のアクセス)
- 揮発性(特にIoT機器)、不揮発性

VFS
ファイルとディレクトリ管理
ディスクキャッシュによる非同期処理

統一した枠組みを提供



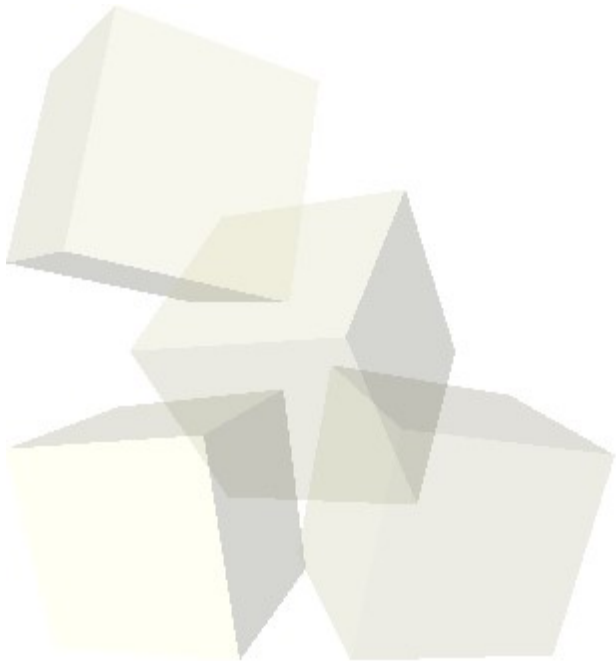
■ 二次記憶の階層化

- アクセス頻度の低いデータを、
より低速だがより安価な記憶メディアに転送
- 直感的な例は、「バックアップデータ」
 - データが破損した場合にのみアクセスされる
 - 書き込みも定期的(1日に1回, 週に1回, など)
 - アクセス頻度: 非常に低



14.4

事例：MS-DOSとUNIXの ファイルシステム





■ FAT (File Allocation Table)

- MS-DOS, Windows95/98
- デジカメ, シリコンオーディオプレーヤ

■ i-node

- 伝統的UNIX系ファイルシステムで用いられる
データ構造

■ FAT

- 基本はインデクス方式
 - 複数セクタを1クラスタとして管理
 - クラスタの総数に上限あり
 - FAT12:4084
 - FAT16:65524
- よって, 1クラスタのサイズはディスク容量をクラスタ数で割ったものになる
- どんな小さなファイルでも1クラスタ消費する

FATにおけるディスク構成

MBR (Master Boot Record) が
格納される場所

ディスクに関する情報、OSを起動する
ために実行するための
や、OSの管理情報
Parameter Block: B

インデクステーブル

FAT1の
バックアップ

ルートディレクトリの
ファイル情報

サブディレクトリや
ファイルの実体

ブートセクタ

FAT1

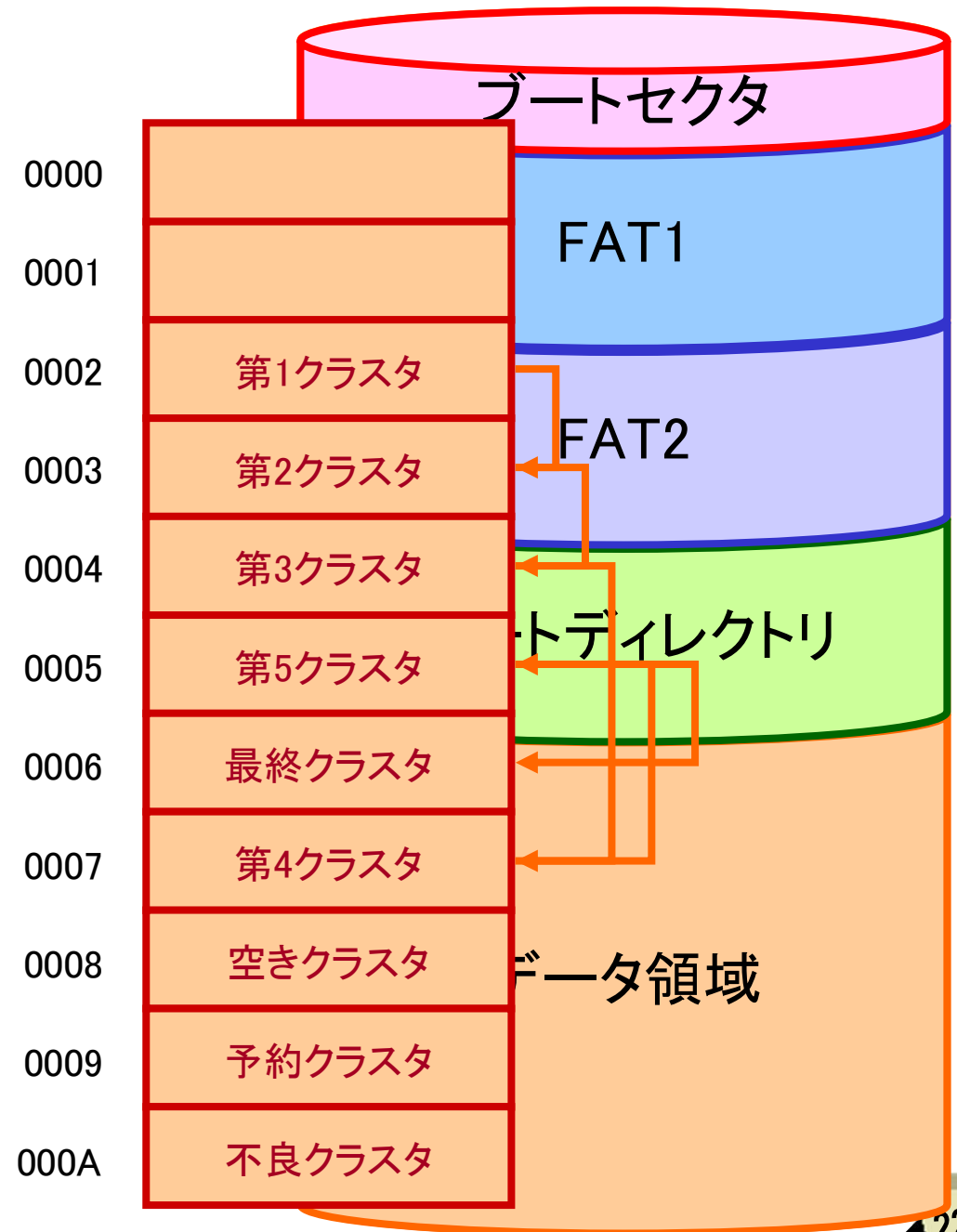
FAT2

ルートディレクトリ

データ領域

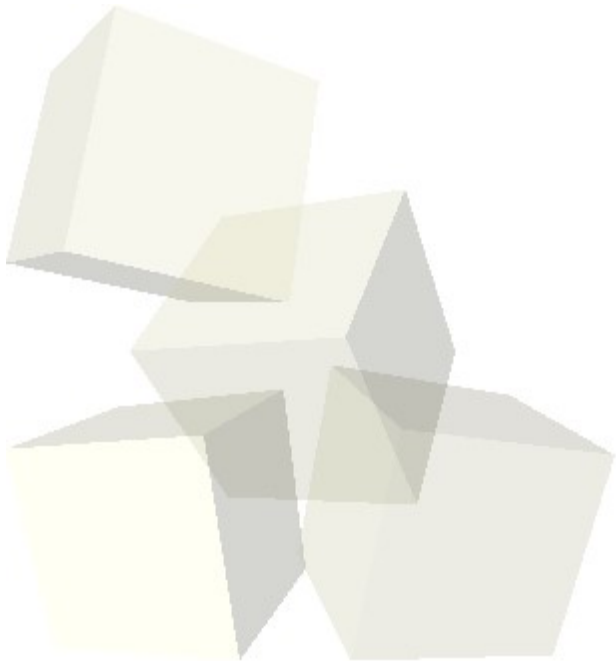
FATにおけるディスク構成

0000	
0001	
0002	0003
0003	0004
0004	0007
0005	0006
0006	FFFF
0007	0005
0008	0000
0009	0001
000A	FFF7



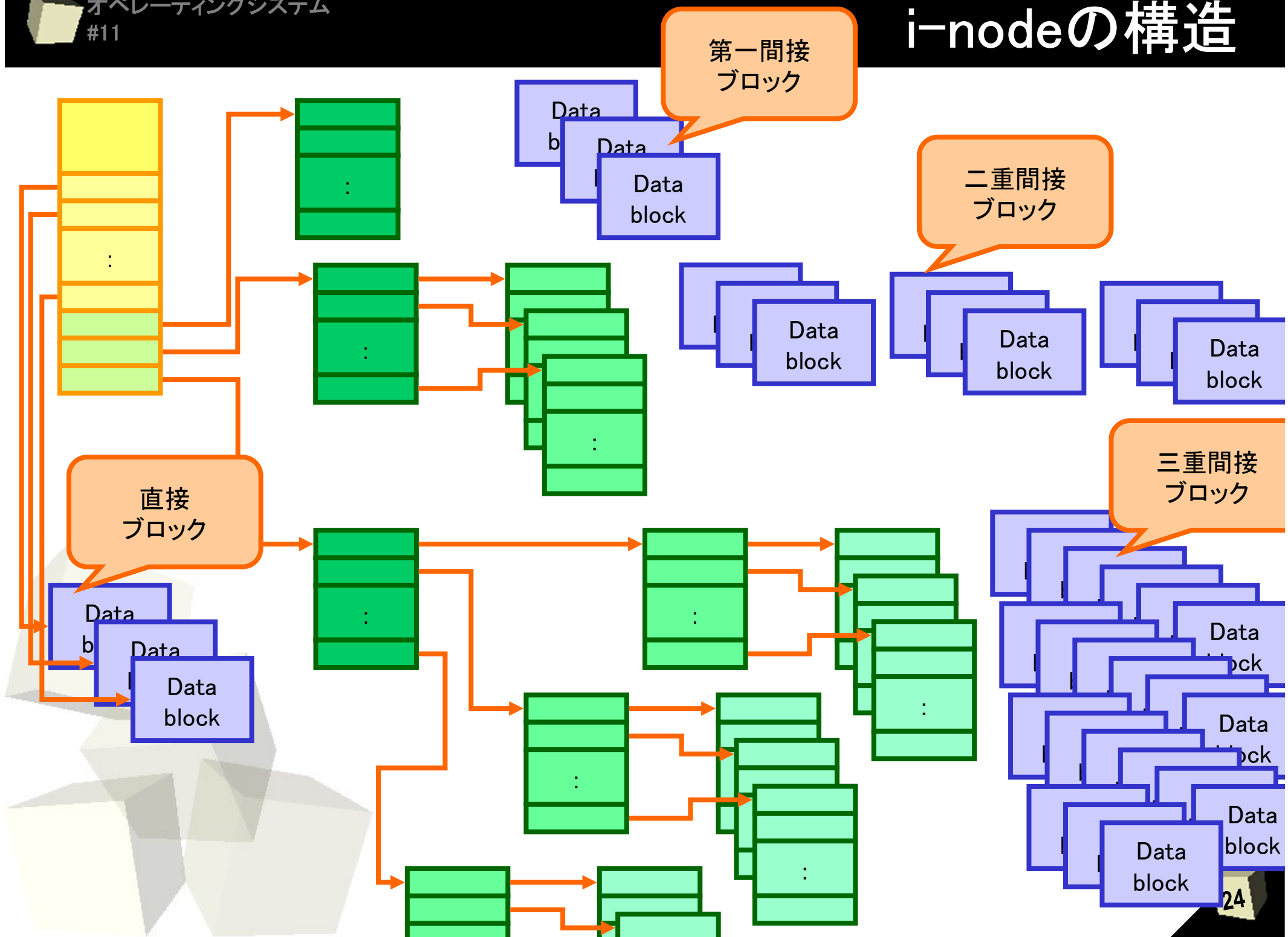
■ i-node

- 基本はインデクス方式
- 階層構造を用いた設計
 - より大容量のディスクを少ないインデクス領域で管理



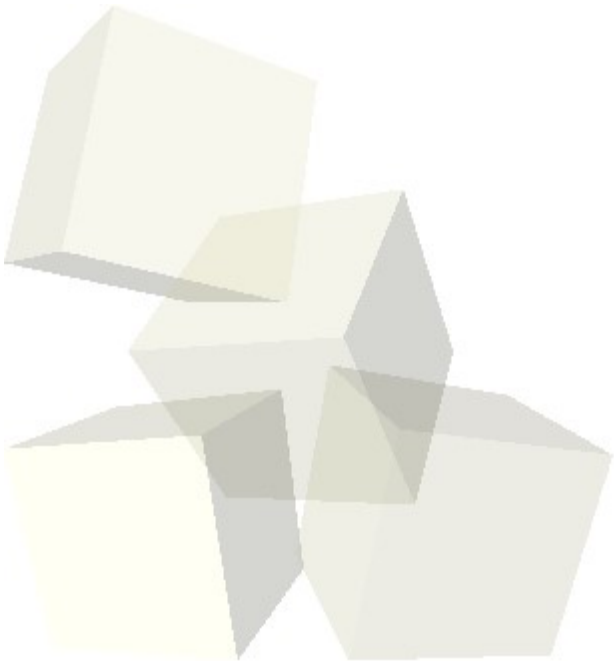


i-nodeの構造



■ 間接方式

- 3重間接では $1024^3 = 1,073,741,824$ クラスタを扱える
 - 1クラスタ = 8kB の場合, 8TB



■ ディスクキャッシュ

- 遅いディスク上のデータの一部を主記憶に置く
- 高速アクセス可能だが、
書き込み時にディスクへの変更部分の反映が必要

• ライトスルー

→ 書き込み時には、主記憶上のイメージだけでなく
同時にディスクに対しても書き込みを行う

• ライトバック

→ 書き込み時には、主記憶上のイメージに対してのみ変更
→ 必要になった時点で変更部分をまとめてディスクに反映

■ 非同期入出力(ノンブロッキングI/O)

- I/Oが必要となった際, プロセスを停止せずに I/O結果に依存しない部分をI/O操作と並行して行う

■ VFS

- 様々な二次記憶へのプログラム側からのアクセス方式を統一

■ ステージング

- ファイルシステムの仮想化手法
- 低速ディスクの内容を一部高速な主記憶に置く
ディスクキャッシュと同様の考え方
- 低速メディア(テープ, 光ディスク)の内容を一部高速な二次記憶(ハードディスク)に置く

■ FAT

- MS-DOSで採用されたファイルシステム
- 基本はインデクス方式
- 複数セクタを1クラスタとして管理
- クラスタの総数に上限あり

■ i-node

- 伝統的UNIX系ファイルシステムで用いられる
- 基本はインデクス方式
- 階層構造を用いた設計
 - より大容量のディスクを少ないインデクス領域で管理