

第6章 ファイルの管理

花田 英輔

(このPowerPointは渡辺名誉教授作成のものを花田が一部改編した)

1

ファイルとは(教科書6.1)

- ▶ 関連する情報をひとまとまりにしたもの
または、そのための入れ物
 - 外部記憶装置の中に保持
 - 名前で識別する論理的概念
(物理的には多数のブロックに**分散して配置**)

名前	更新日時	種類	サイズ
ディレクトリ (フォルダ)			
WindowsMobile	2012/03/02 17:59	ファイル フォル...	
winsxs	2013/11/14 9:08	ファイル フォル...	
ファイル			
_default	2009/06/11 6:42	MS-DOS プログ...	1 KB
atiogl.xml	2009/06/17 7:53	XML ドキュメント	18 KB
ativpsrm.bin	2012/03/01 15:48	BIN ファイル	0 KB
bfsvc.exe	2010/11/20 21:16	アプリケーション	64 KB
bootstat.dat	2013/11/19 9:13	DAT ファイル	66 KB
DirectX.log	2012/03/01 16:08	テキストドキュ...	86 KB
Dtinstall.log	2012/03/01 16:08	テキストドキュ...	86 KB
epplauncher.mif	2013/10/16 18:34	MIF ファイル	2 KB
explorer.exe	2011/02/25 14:30	アプリケーション	2,555 KB
fiveupdata.exe	2009/07/14 10:14	アプリケーション	14 KB

2

ファイル拡張子

- ▶ Windows⇒ファイル名の末尾に種別を示す文字列
＝拡張子

winsxs	2013/11/14 9:08	ファイル フォル...	1 KB
_default	2009/06/11 6:42	MS-DOS プログ...	1 KB
atiogl.xml	2009/06/17 7:53	XML ドキュメント	18 KB
ativpsrm.bin	2012/03/01 15:48	BIN ファイル	0 KB
bfsvc.exe	2010/11/20 21:16	アプリケーション	64 KB
bootstat.dat	2013/11/19 9:13	DAT ファイル	66 KB
DirectX.log	2012/03/01 16:08	テキストドキュ...	86 KB

- 拡張子を表示するには、
コントロールパネル>デスクトップのカスタマイズ
>フォルダーオプション>表示>登録されている
拡張子は表示しない>チェックを外す

やってみよう！

3

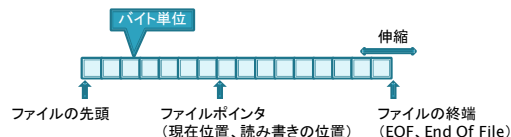
ファイルの編成(教科書6.2)

- ▶ 順編成
 - 先頭から順次アクセス
- ▶ 直接編成
 - 位置を指定してランダムアクセス
- ▶ 索引順編成
 - 各々のレコードにキー保持、キーによるアクセス可能

4

UNIXのファイル編成

- ▶ 1バイト単位の順編成を基本



- ▶ 各種入出力装置もファイル概念に統一
 - キーボード、スクリーン、プリンタ、...
 - 「ブロックデバイス」: ブロック単位で読み書き(HDD等)
 - 「キャラクタデバイス」: キャラクタ単位で読み書き(KBD等)

5

ブロックデバイスとキャラクタデバイス (Wikipediaより)

- ▶ ブロックデバイス
 - ブロック形式でデータをやり取りする機器
 - ハードディスクドライブ/CD-ROMドライブ/メモリ領域などのアドレス指定可能な機器
 - ランダムアクセスとシークが可能ことが多い
 - 一般にバッファを使った入出力を行う
- ▶ キャラクタデバイス
 - システムが一字ずつデータを転送する機器
 - テレタイプ端末、モデム、仮想コンソール、擬似端末などのようなバイトストリーム型機器
 - ランダムアクセスはサポートしていないのが普通
 - システムは文字単位に逐次的に読み書きを行う

6

ファイルの操作(教科書6.3)

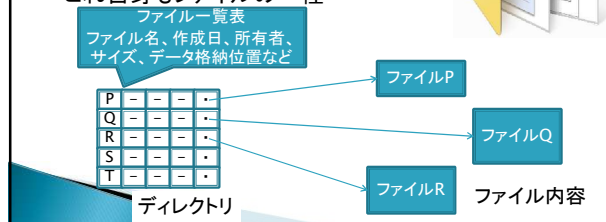
- ▶ ファイルを操作するためのOS機能
 - 応用プログラムが依頼、OSが実行

ファイルの生成(create)
 ファイルの削除(delete)
 ファイルのオープン(open)
 データの入力(read)
 データの出力(write)
 ファイルのクローズ(close)
 ファイル名の変更(rename)
 ...

7

ディレクトリ(教科書6.4)

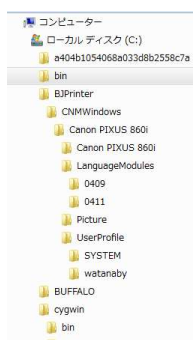
- ▶ 見え方=ファイルを納める入れ物
 - Windows, MacOSでは「フォルダ」と言う
- ▶ 実際の構成=ファイルの管理簿
 - ファイルとは別に作成、ファイルとリンク
 - これ自身もファイルの一種



8

階層型ディレクトリ

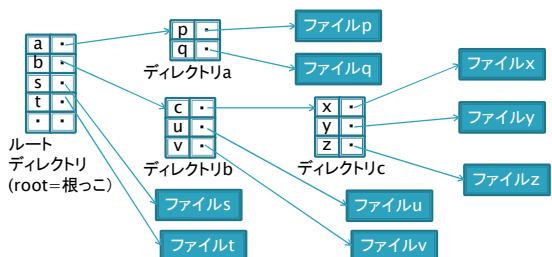
- ▶ ファイル群をフラットに配置
=単一レベルディレクトリ
- ↓
- ▶ ユーザごとにファイル群を
区分して配置
=2レベルディレクトリ
- ↓
- ▶ 多段階に区分して配置
=階層型ディレクトリ



自分のPCはどうか
見てみよう！

9

階層型ディレクトリ



絶対パス指定(ルートからの経路): `/b/c/y`
 /(=root)から記述、Windowsでは「/」の代わりに「¥」
相対パス指定(作業ディレクトリからの経路): `./c/y`
 現在の作業ディレクトリから順にディレクトリ名を記述

10

ディレクトリの操作(教科書6.5)

操作内容の例)

- ▶ 通常ユーザ向け
 - ディレクトリの作成(mkdir, make directory)
 - ディレクトリの削除(rmdir, remove directory)
 - ディレクトリ内のファイル一覧(ls, list)
 - 作業ディレクトリの移動(cd, change directory)
- ▶ システム管理者向け
 - 部分木をつなぐ(mount)
 - 部分木を切り離す(umount, unmount)

11

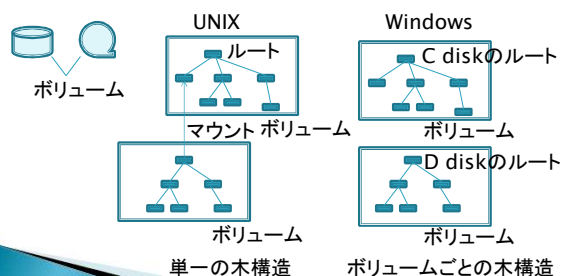
ディレクトリの操作(続)

- ▶ 現在、作業の対象となっているディレクトリ
= **作業ディレクトリ**(Working Directory)
または **カレントディレクトリ**(Current Directory)
- ▶ ディレクトリ移動に関する操作
 - `cd c/y`: カレントディレクトリを、今いるところの下のcディレクトリの下のyディレクトリへ移動せよ(change directory)
 - `cd ..`: カレントディレクトリを一つ上へ移動せよ
 - `cd /`: カレントディレクトリをroot(トップ)に移動せよ

12

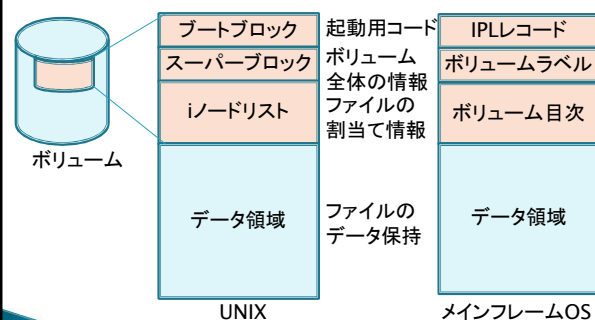
ファイルシステムとボリューム(教科書6.6)

- ▶ **ファイルシステム** = ファイルを操作するOS機能
- ▶ **ボリューム** = 1記憶媒体(ファイル蓄積場所の単位)



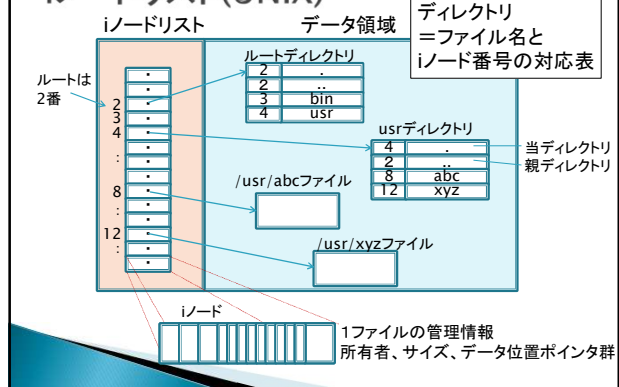
13

ディスクボリュームの構成



14

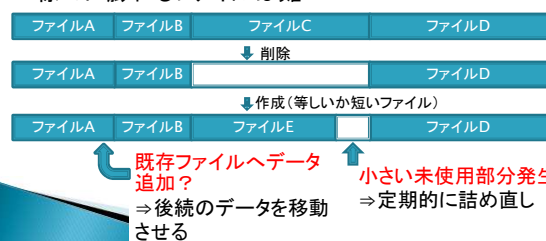
iノードリスト(UNIX)



15

ファイルのデータ領域の割当て

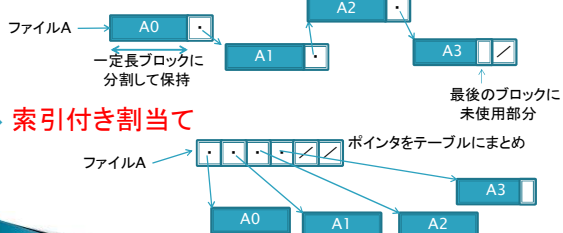
- ▶ 大きいファイルも小さいファイルも効率的に保持する工夫
- ▶ **連続割当て** = ディスクの先頭から詰めて格納
 - 削除・追加繰り返して小さい穴が分散
 - 徐々に膨れるファイルは難



16

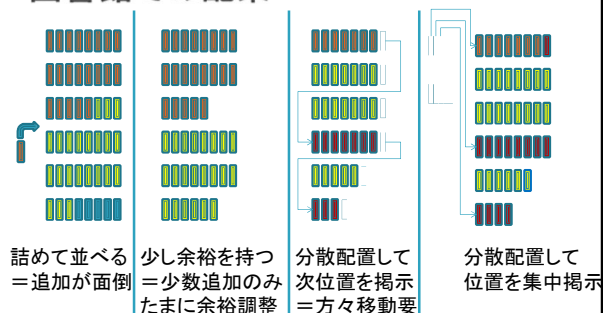
ファイルのデータ領域の割当て

- ▶ **鎖状割当て**
 - リンクポインタ領域必要 (= ブロック長が中途半端に)
 - ダイレクトアクセス難 (= N番目アクセスはN回リンク辿る)



17

図書館での配架

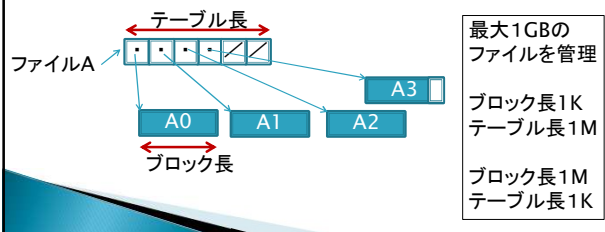


18

索引付き割当ての問題点

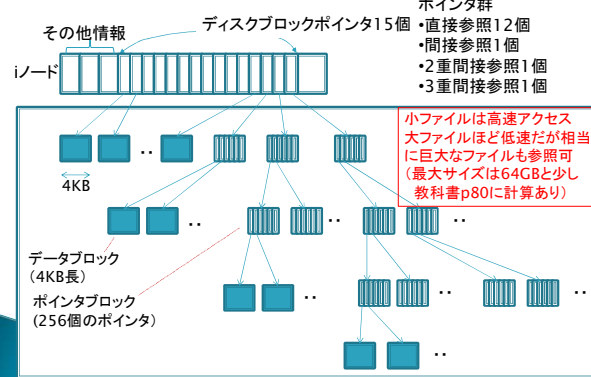
▶ テーブル長とブロック長をどうする？

- ブロック長を長くすると、ブロック中の無駄増
- テーブル長を長くすると、テーブル中の無駄増
- 両方とも短くすると、ファイル最大長が減少



19

UNIXにおける方法(iノード)



20

今回の課題

1. 1つのファイルとして見えるデータは、記憶装置の中で連続した領域に保持せず、分散して保持することが多い。その理由を説明せよ。
2. 「ディレクトリ」の内容と役目を記せ。
3. (予習)「プロセス」とは何か、調べて記せ。

- ▶ 今回のファイル名は“学籍番号-OS07.docx”
(例: 24238000-OS07.docx)としてください
- ▶ 締切: 11月21日(金) 18:00 (遅れた場合は減点)

記載時の注意事項

- ▶ 参考資料(Webページ)がある場合は**出典を書くこと**
 - **出典を書かずに引用した場合は減点対象です**

21