

オペレーティングシステム(OS)

柴山 潔

13. ファイルシステム (1)

- ファイルとファイル装置
- ファイルの管理と操作
- ファイル構造とファイルアクセス方式(1)

ファイルとファイル装置 —概要—

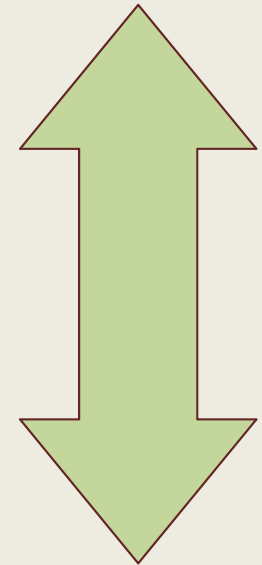
[定義3.7] **ファイル(file)**

- 広義のプログラム（命令やデータ）を格納/保持するための名前付けた論理的単位

(格納)

- **ファイル装置**: ファイルを実際に格納する物理的なハードウェア装置

[ソフトウェア]



[ハードウェア]

ファイル (1)

- コンピュータシステムのユーザ(実際には, OSやユーザプログラム)が作成
 - 各々が型(タイプ(type))と論理的構造(=ファイル構造)をもつ
- プログラム(OSやユーザプログラム)が実行時にファイルに対して行う処理(=ファイル操作)機能
 - (具体的) (1) 生成; (2) 消去; (3) 併合; (4) 分割; (5) 編集(更新/変更); (6) 転送(移動/移送); など

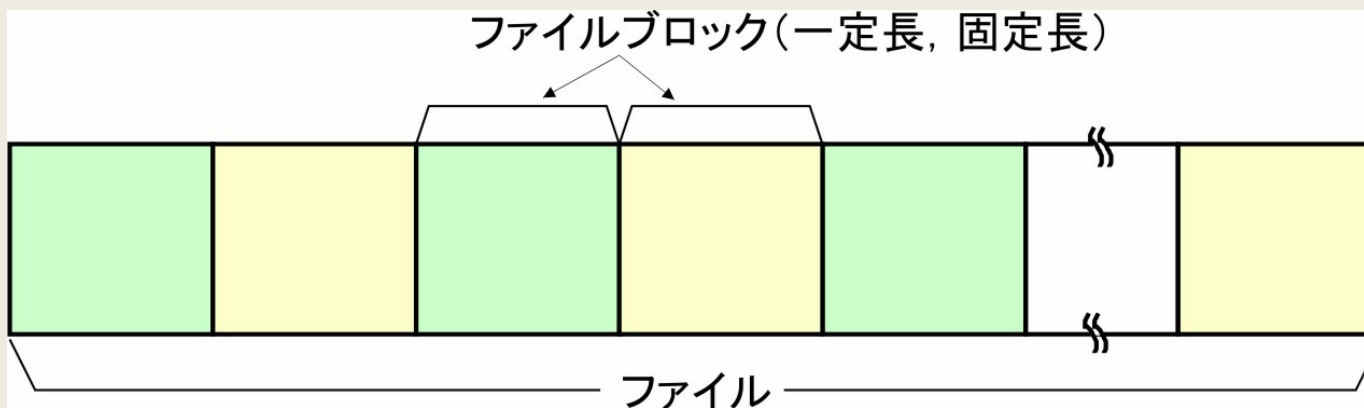
- ファイル操作に対して満たすべきファイルの要件
 - ユーザプログラムによるファイル操作の対象
 - ユーザプログラムによるファイル操作が可能

ファイル (2)

■ ファイルの代表例

- (a) ソースプログラム: ユーザが直接作成・編集
- (b) オブジェクトプログラム: コンパイラが生成 → 実行時にプロセス領域(コード)としてプロセス割り付け
- (c) 実行可能プログラム: コンパイラが生成
- (d) データ(広義): ユーザやユーザプログラムが直接作成・編集 → 実行時にプロセス領域(データ)としてプロセス割り付け

- ファイルを構成する最小単位は固定&一定長ブロック(=ファイルブロック(file block))



ファイル装置（再考）

- 内部装置（本体，プロセッサ-メインメモリ対）としてではなく，外部装置として装備するメモリ（ハードウェア）
＝ “補助メモリ（外部メモリ，2次メモリ）”， “ストレージ(storage)”
← “メインメモリ（内部メモリ，1次メモリ）”に對比

[役割]

- コンピュータのユーザがプログラムやデータを“ファイル”として格納
 - ユーザがOSの管理下でファイル装置に格納してあるファイルを直接編集・使用
- プロセッサからは，メインメモリと相対的比較して，アクセス速度よりも格納機能（容量）を重視したメモリ階層という位置付け
＝ メインメモリのバックアップメモリ＝補助メモリ
 - メインメモリ機能の空間的改善を図るために利用するメモリ階層
＝ 仮想メモリ:ファイル装置はメインメモリ(実メモリ)のバックアップメモリ
 - プロセッサからはメインメモリを経由する間接アクセス

- 種々のファイル装置がメモリ階層(＝ファイル装置階層)を構成
(例) ハードディスクドライブ装置;磁気テープ装置;フラッシュメモリ;光ディスクドライブ装置;光磁気ディスクドライブ装置;など

ファイルシステム(file system)

= OSのファイル管理機能(システムサービス)

- コンピュータシステムのユーザ(広義, 人間の他にOS自身を含むすべてのプログラムを含む)に対して, ファイルに対する統一的な/一元化した管理機能を提供

[具体的な機能]

1. ファイルとファイル装置(の領域)との対応付け(マッピング)
 - ファイルのファイル装置(の領域)上への割り付け(=ファイル割り付け)
= ファイル装置での(メモリ)領域管理
2. ファイルに対する簡便な構造化方式&アクセス方式(→ユーザインタフェースの一部を構成)を統一して, ユーザに提供
3. ユーザやユーザプログラムによる不正アクセスからファイルを保護(=ファイル保護)

ファイル操作

- **ファイルに対する統一的な/一元化した操作機能** ← **ファイルシステムが提供**

- (1) **オープン**(open): 「プロセス(プログラム)による**ファイル使用の開始**」の通知を受け付け → 当該ファイルに対する管理機能を**開始**
- (2) **クローズ**(close): 「プロセス(プログラム)による**ファイル使用の終了**」の通知を受け付け → 当該ファイルに対する管理機能を**終了**
- (3) **生成**: ファイル装置上でファイル格納のために**領域を確保**(+ファイルの格納)
- (4) **消去**(削除): ファイル装置上でファイル格納のために確保していた**領域を解放**
- (5) **書き込み**(ライト(write)/**更新**): (メインメモリから)**ファイル装置へ**ファイル(内容)を**書き込む**
- (6) **読み出し**(リード(read)): **ファイル装置から**(メインメモリへ)ファイル(内容)を**読み出す**
- (7) **コピー**(複写): ファイルの**コピー(複製)**を作成
- (8) **移動**: ファイル装置上でファイルの格納領域(場所)を**変更**
- (9) **リスト**(list): ファイル属性(例: 名称/サイズ/アクセス日時など)の**一覧表を出力**

➤ **ファイルシステム**の主要な機能として、個別のシステムサービス(モジュール)によって実現

ファイル制御ブロック(FCB) (1)

[定義3.8] **ファイル制御ブロック(FCB; File Control Block)**

- **ファイル管理のために実行時(動的)に生成するファイルに関する情報(=ファイル情報)**
= **ファイル記述子(ファイルディスクリプタ(file descriptor))**

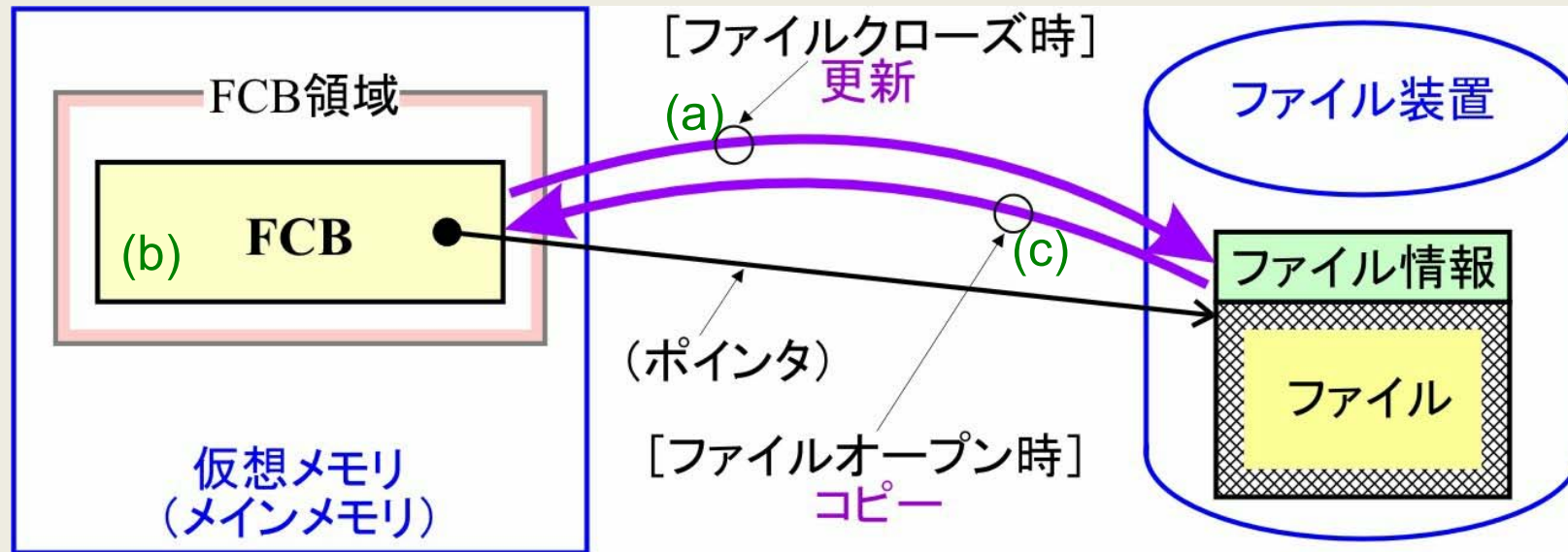
➤ **OS(ファイルシステム)が管理**

➤ **FCB**はPCB(プロセス制御ブロック)の**ファイル版**

➤ **ファイル操作**((1)~(9))機能のほとんどは、実際には、「**仮想メモリ(実際には、メインメモリ)上にあるFCBに対する操作**」

- **FCB**操作による(ファイルそのものの)管理・制御:(1) **オープン**; (2) **クローズ**; (3) **生成**; (4) **消去**; (9) **リスト**;
- **FCB**操作&実際のファイル装置上の(ファイルへの直接のアクセスによる)ファイル操作:(5) **書き込み**; (6) **読み出し**; (7) **コピー**; (8) **移動**;

ファイル制御ブロック(FCB) (2)(図説)



- **FCB**の元となる静的なファイル情報は、ファイルごとに、ファイル内容のヘッダ (header; 先頭部分の見出し情報)として、ファイル装置に格納
- (a) ファイルの**オープン**時に、ファイル装置上のファイル情報を**仮想メモリ**(実際は、**メインメモリ**)上の**FCB領域**に**コピー**; (b) ファイル使用中は、その**FCB領域**で、**FCB**として保持; (c) ファイルの**クローズ**とともに、ファイル装置上のファイル情報として、**仮想メモリ**(実際は、**メインメモリ**)上の**FCB領域**から書き戻し(**更新**)

ファイル情報(例)

= FCBに設定するファイル情報の項目(例)

- (1) **名前**: ユーザが付けた名前(論理名), = **ファイル名**
- (2) **型**: ファイル内容の種類
(例) ソースプログラム/オブジェクトプログラム/実行可能プログラム/データ
- (3) **サイズ**: ファイルの占有領域の大きさ ← **ブロック総数**
- (4) **構造に関する情報**: **ファイル**割り付け方式の別
(例) FAT / UFS / NTFS
- (5) **保護に関する情報**: アクセス権の別や当該権限を有する者の識別, = **属性**
(例) **読み出し可能**(Readable)/**書き込み可能**(Writable)/**実行可能**(eXecutable), **所有者**(owner; 作成者)/**使用者**(user; ユーザ)
- (6) **参照局所性に関する情報** ← 時間情報や統計情報
(例) **時間情報**: 作成日時;(最終)更新日時;(最終)アクセス日時, **読み出し/書き込み回数**
- (7) 格納する/対応付けた**ファイル装置上の物理アドレス**(= **ポインタ**) = ボリューム(volume)上の物理アドレス)

(必須!) 対応付け
(ファイル割り付け)

ファイルのオープンとクローズ

- プロセス（プログラム）自身による、実行中に使用するファイルについての、OS（ファイルシステム）へ「ファイル使用の開始（←**オープン**）/終了（←**クローズ**）」の通知/宣言

■ **オープン**操作手順

1. OSが仮想メモリ（実際には、メインメモリ）上にFCB領域を確保
2. 使用**ファイル（名）を特定（識別）**
3. そのヘッダ部にあるファイル情報を、FCB項目として、FCB領域に**設定&コピー**

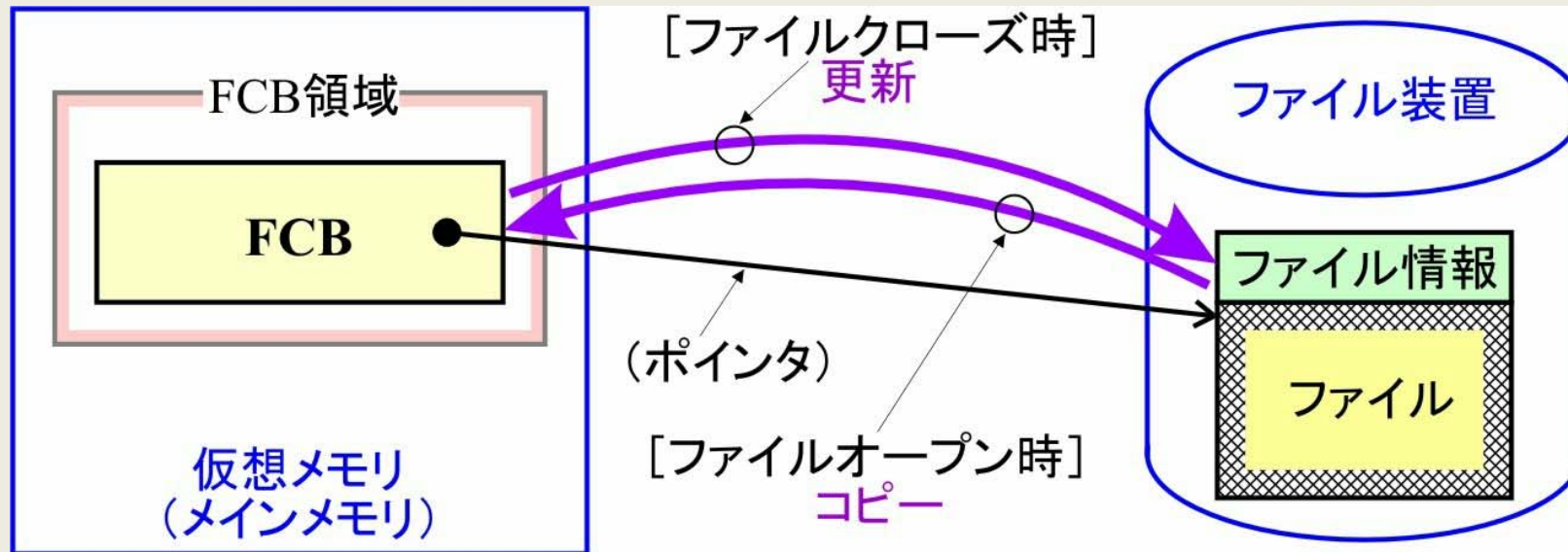
■ **クローズ**操作手順

1. FCB領域で更新された項目によって、当該ファイルのヘッダ部にある**ファイル情報そのものを更新**
2. 仮想メモリ（実際には、メインメモリ）上のFCB領域を**解放（=OSに返却）**

オープン操作 → （種々の/一連の）**ファイル操作** → **クローズ**操作

（例）読み出し、書き込み（更新）など

ファイルのオープンとクローズ(図説)(再掲)



ファイル構造とファイルアクセス方式 (1)

[定義3.9] **ファイル構造**

- ファイルの論理的な構造 (論理構造) = ファイル編成
 - ファイルシステムが規定

- OS (ファイルシステム) は, ユーザプロセス (ユーザプログラム) によるファイル操作のために, 「一元的/統一したファイル構造」を提供
 - = ファイルシステムは, それが規定するファイル構造に対する「一元的/統一したファイル操作」を実現
- ユーザプロセス (ユーザプログラム) は, ファイルシステムを介して (=ファイルシステムが提供する論理的なファイル操作機能によって), それらのファイルに間接的にアクセス

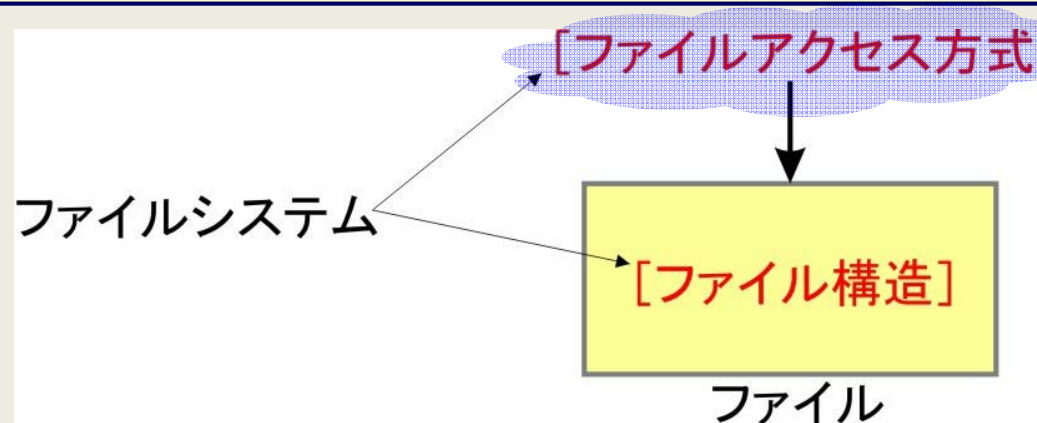
ファイル構造とファイルアクセス方式 (2)

[定義3.10] ファイルアクセス(file access)方式

- ファイル構造(論理構造)にしたがうファイルへのアクセス(=読み出し+書き込み)方式
 - ファイルシステムが規定

- ファイルシステムの機能として, ファイルシステムが規定するファイル構造にしたがって実現/実装

- ファイルシステムの主要な機能として, “ファイルアクセス方式”に“ファイル構造”を含めることも



ファイルアクセス方式(ファイル構造を含む) (1) —目的—

- 「ファイルそのもの」&「ファイルへのアクセス」の“透明化”

(具体的)

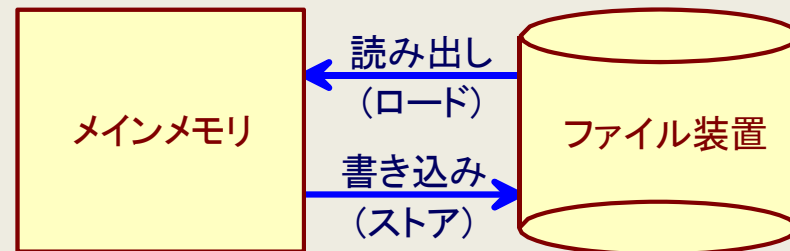
- ファイルへのアクセス主体であるプログラム/プロセスが、ファイル装置に格納してあるファイルの場所や格納形式(←物理構造, =ファイル割り付け)をまったく意識することなく、ファイル構造(←論理構造)にアクセス可能
- ファイルアクセス方式は「ファイル構造に対するファイル操作の一元化/統一」によって「ファイルアクセスの“透明化”」を実現

ファイルアクセス方式(ファイル構造を含む) (2) —分類—

- (A) **逐次アクセス**: ファイルを構成するブロックの先頭から順に(並び順で)アクセス
- (B) **直接アクセス**: ファイルを構成するブロックのどれにも任意順(=順不同)でアクセス可
= ランダムアクセス(random access; 任意順アクセス)

■ 実際のファイルアクセス —ファイル装置に対するアクセス方向による分類—

- (a) 読み出し: [ファイル装置→メインメモリ]転送操作 =ロード(load)
- (b) 書き込み: [メインメモリ→ファイル装置]転送操作 =ストア(store)



ファイル構造 —代表例—

(A) 逐次アクセスファイル

- ファイルシステムがあらかじめ規定する順序(生成順が代表的)でファイルブロックを並べる

(B) 直接アクセスファイル

- 番号付けしたファイルブロックの集まり

(C) インデックス付きファイル(indexed file)

- インデックス(index; 索引)によって直接アクセス(→直接アクセスファイルの発展形)

逐次アクセスファイル (1)

- ファイルシステムがあらかじめ規定する順序(生成順が代表的)でファイルブロックを並べる



- ファイルアクセス方式 → 逐次アクセス(A)が普通
 - その場合のアクセス順は, ファイルブロックの並び順そのままであり, 変更不可(=固定)

■ **逐次アクセスファイル装置**: ファイルブロックをファイル装置上の連続する領域(アドレス)に格納しておき, それらのファイルに逐次的に(=並び順=アドレス順)でアクセス, 代表例: 磁気テープ装置

- 「**逐次アクセスファイル装置**に格納するファイルのファイル構造は逐次アクセスファイルに限る」のが普通

逐次アクセスファイル (2) —特徴—

長所

- 連続ブロックの逐次アクセス(読み出し&書き込み)は高速
- 読み出し/書き込みに限ると、ファイル操作は単純

短所

- 読み出し/書き込み以外のファイル操作(例:生成/追加, 部分削除など)は面倒
- ファイル途中でのブロックの挿入や削除が困難

直接アクセスファイル (1)

- 番号付けしたファイルブロックの集まり

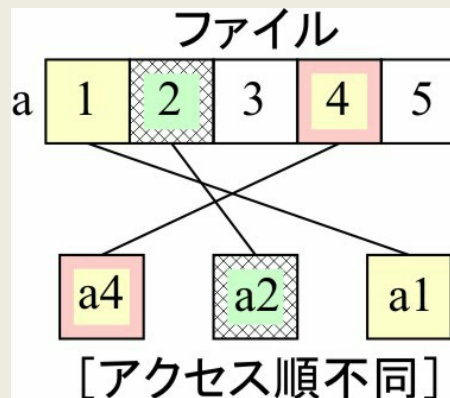
- ブロック番号(相対ブロック番号 → 定義3.11 ↓)を指定すれば, 任意のファイルブロックに直接アクセス(=ランダムアクセス, (B)のファイルアクセス方式)可能

= **ランダムアクセスファイル**(random access file)

[定義3.11] **相対ブロック番号**

- ファイルアクセス方式(ファイル構造を含む)における「当該ブロックの『先頭ブロックからの相対的な位置』(=オフセット(offset))」

- ファイルブロックへのアクセス時に, 相対ブロック番号によって当該ブロックを指定



直接アクセスファイル (2)

- **直接アクセスファイル装置**: どのファイルブロックにも**直接アクセス可能**なファイル装置, 代表例: **ハードディスクドライブ装置**, 例: 呼称に「**ドライブ**」が付くファイル装置 (外に, **光ディスクドライブ装置**など); **フラッシュメモリ** = **ランダムアクセスファイル装置**
 - **直接アクセスファイル装置**の (物理的な) 構造は**直接アクセスファイル**のファイル構造 (論理構造)と同じ → **直接アクセスファイル**と**直接アクセスファイル装置**とのマッピングは容易

- **直接アクセスファイル装置**に格納するファイルの**ファイル構造**のほとんどは**直接アクセスファイル**

直接アクセスファイル (3) —特徴—

長所

- ブロックのアドレスを指定して, そのブロックに直接アクセス可 → 特定ブロックへのアクセスは高速
- ファイル装置への格納方法はOS(ファイルシステム)ごとの自由な設定が可能

短所

- OS(ファイルシステム)ごとの管理が必須&複雑