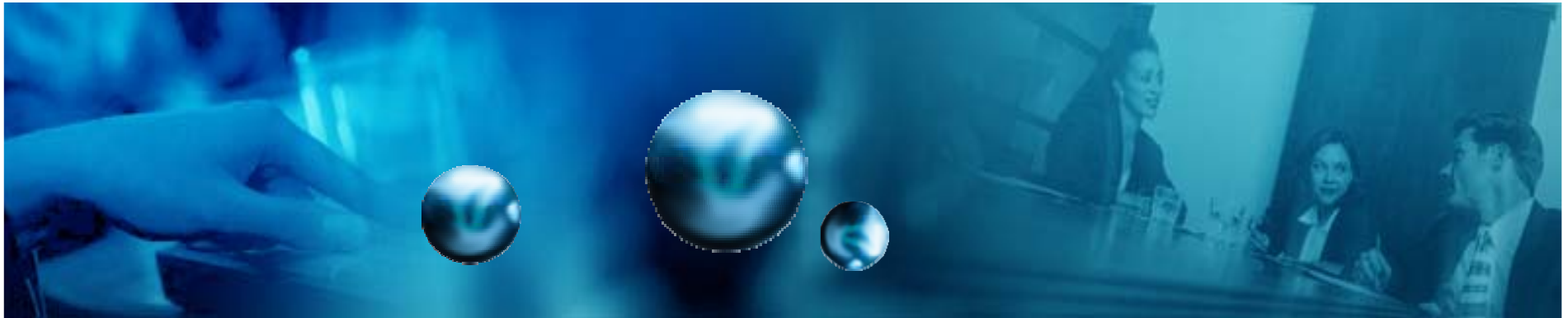


オペレーティングシステム

3章 メモリ管理

3.3 ファイル管理—ファイルシステム—(前半)



大阪大学大学院情報科学研究科
村田正幸

murata@ist.osaka-u.ac.jp

<http://www.anarg.jp/>



3.3 ファイル管理—ファイルシステム—

- ファイル管理
 - メモリとしてのファイル装置における領域管理すなわちメモリ管理機能
- 論理的情報の格納単位であるファイルを、ファイル装置というメモリ領域上で管理する機能
 - ファイルシステムと呼ぶシステムサービスを担うOSモジュールによって実現する
- OS機能としてのファイルシステムの原理
 - ユーザプログラムからハードウェアであるファイル装置を隠ぺいし、さらには、ファイル装置へのファイルの格納形式およびユーザプログラムによるファイルの操作方式を一元化する



3.3 ファイル管理ーファイルシステムー

3.3.1 ファイルとファイル装置

[1] ファイルとファイル装置ー概要ー

- ハードウェア装置としてのファイル装置と、それに格納するソフトウェアとしてのファイルとの違い？

定義3.7(ファイル)

広義のプログラム(命令やデータ)を格納あるいは保持するための名前付けた論理的単位をファイルという

- ファイル装置
 - ー アクセス速度よりも容量を重視するメモリ階層
 - ー メインメモリと比較すると、アクセス速度では劣るが容量では優る



3.3 ファイル管理ーファイルシステムー

3.3.1 ファイルとファイル装置

[2] ファイル

- コンピュータシステムのユーザ(OS、ユーザプログラム)が作成するソフトウェア
 - 型(タイプ)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)機能は、
 - (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)
- ファイルの代表例
 - (a) ソースプログラム:ユーザが直接作成し編集する
 - (b) オブジェクトプログラム:コンパイラが生成する
 - (c) 実行可能プログラム:コンパイラが生成する
 - (d) データ(広義):ユーザやユーザプログラムが生成し編集する
- ファイルを構成する最小単位
 - 固定すなわち一定長ブロック(ファイルブロック)←1KBとか4KBとか

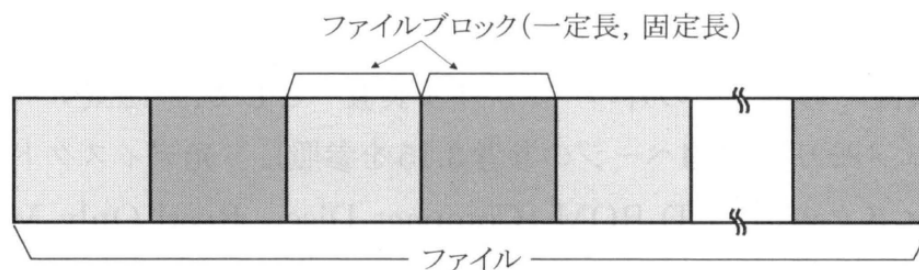


図 3.19 ファイルの構成単位



3.3 ファイル管理—ファイルシステム—

3.3.1 ファイルとファイル装置

[3] ファイル装置

- 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)
- 内部装置であるメインメモリ(内部メモリ、1次メモリ)に対して、補助メモリ(外部メモリ、2次メモリ)と呼ぶ
- 情報の格納機能を強調する場合は「ストレージ」
- ファイル装置の役割
 - 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する。
ユーザは、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に編集さらには使用する
 - プロセッサからは、メインメモリに比較すると、アクセス速度よりも格納機能、すなわち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あるいは補助メモリになる
 - 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として、仮想メモリ機構においても、メインメモリ(実メモリ)のバックアップメモリの役割
 - プロセッサからは、直接アクセスではなく、メインメモリを経由する間接アクセスとなる



3.3 ファイル管理—ファイルシステム—

3.3.1 ファイルとファイル装置

[3] ファイル装置

- ファイル装置の種類
 - アクセス速度と容量(コスト)のトレード

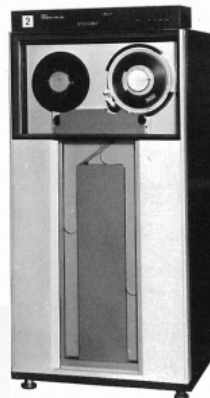
ハードディスクドライブ装置→HDD



フラッシュメモリ→SSD



光ディスクドライブ装置→CD-ROM



磁気テープ装置



光磁気ディスクドライブ装置→MO



Mr. Murata

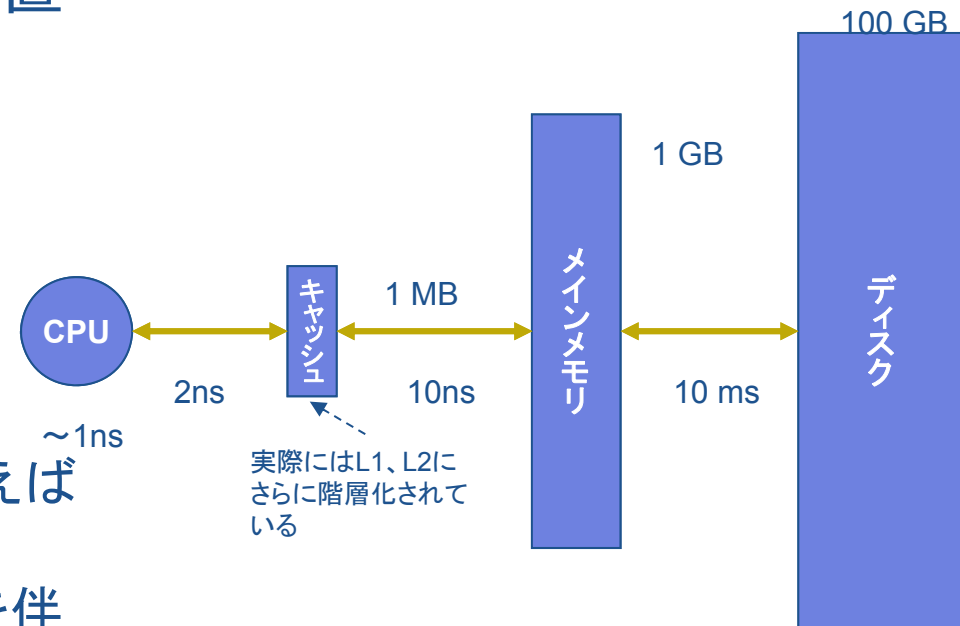
DVD (Digital Video Disc)





ファイル装置の役割

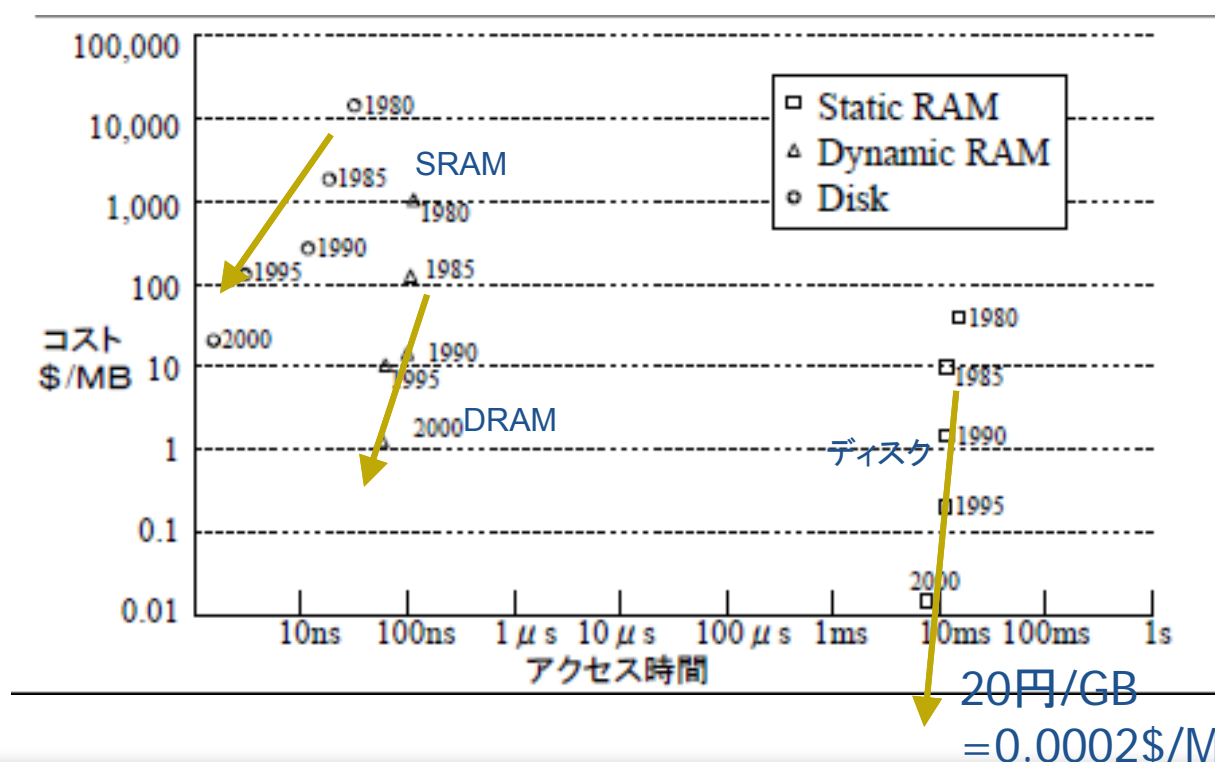
- キャッシュ(速いが少量)、メモリ(中間)、ファイルシステム(大容量だが遅い)の3つの記憶装置がある
 - 階層構成をとる:ごく自然
- ファイル装置
 - 最後の砦
 - 効率の面
 - 信頼性の面
 - 遅さをどうやってごまかして使えばよいか?
 - 特にディスクは機械的な動作を伴うため、メモリとは異なる使い方がある





ディスクとメモリのコスト推移

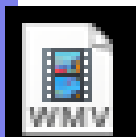
- ディスクの発展はメモリの発展に追随している
 - アーキテクチャへの変化を促さない





ハードディスク

ハードディスク
の構造

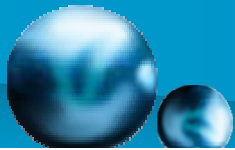


A1_K004.wmv

•http://www2.iwate-ed.jp/joho/A/A1_K004S.html

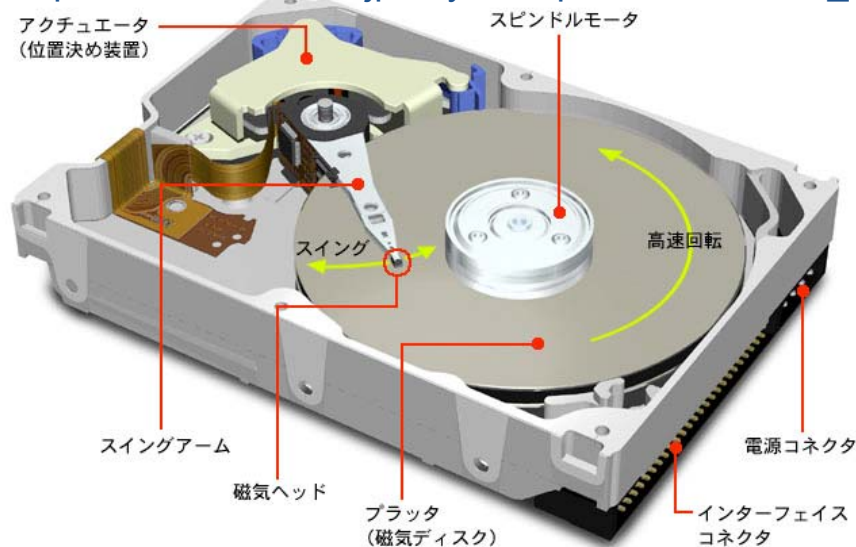


M. Murata

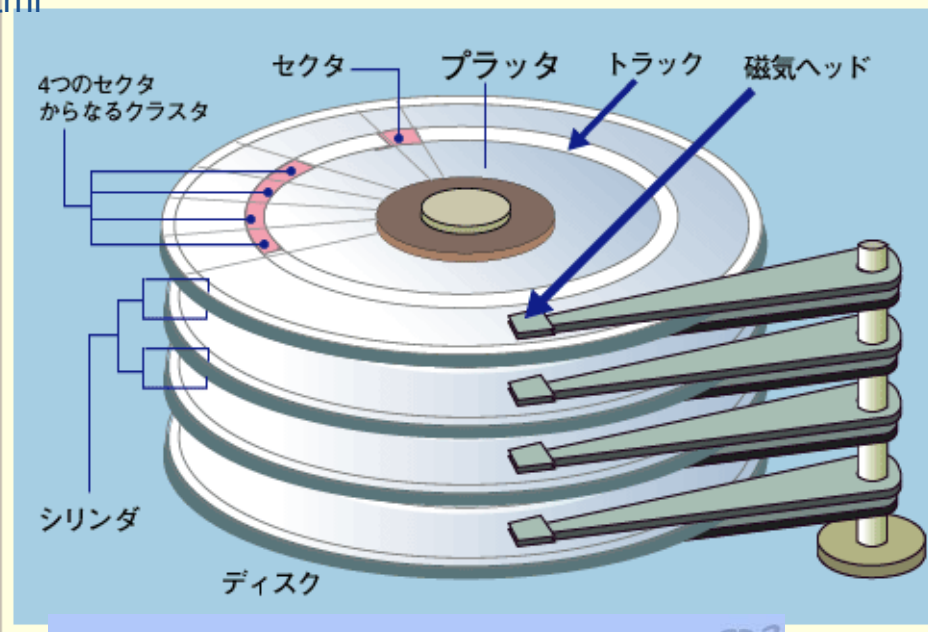


ハードディスクの構造

http://www.infonet.co.jp/ueyama/ip/hardware/hard_disk.html



の

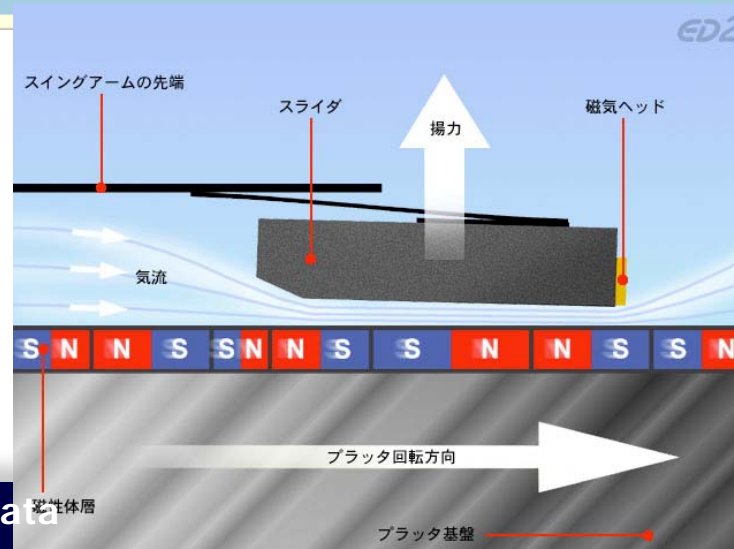


ヘッドの浮上量
0.01 μm

煙草の煙
1~3 μm

指紋の断面
3~5 μm

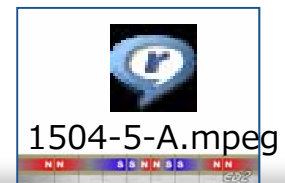
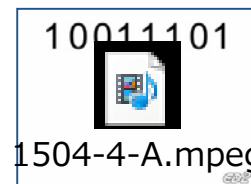
髪の毛の断面
80~120 μm





ハードディスクの仕組み

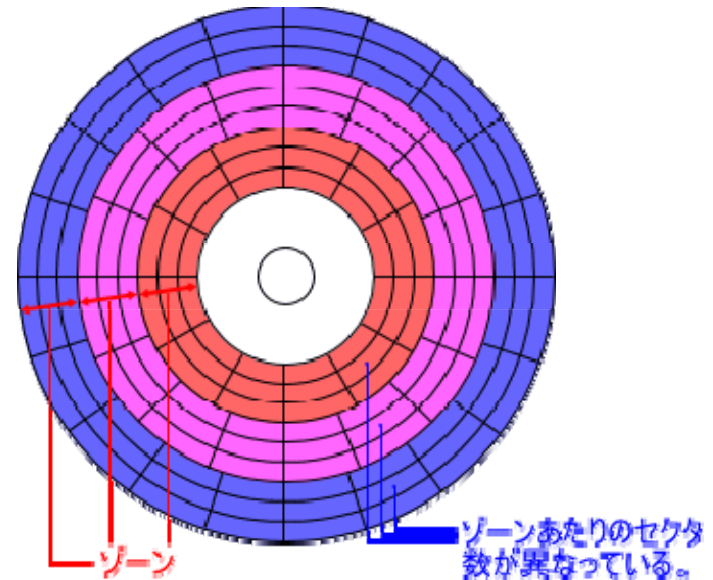
- <http://www.sugilab.net/jk/joho-kiki/index.html>
- ハードディスクドライブの動き
 - 情報を記録するプラッタ(磁気ディスク)は、1台のドライブの中に数枚内蔵されていて、磁気ヘッドは、1枚のプラッタにつき2つ、両面に記憶できるようにつけられている
- 書き込みのしくみ
 - コイルを流れる電流により、コアが磁化され、ディスクの表面の磁性体が磁化される。この原理を用いて電流の向きを変えることでディスクの表面に磁化パターンを記録する
- 読み込みのしくみ
 - ディスク表面の磁化パターンにより、コアが磁化されようとする。ディスクが回転すると、コアに生じた磁界を打ち消すように電流が流れる
- 書き込み信号処理
 - 記録したい情報を0を低い電圧、1を高い電圧で表して、ヘッドの読み書き信号に変換する。この書き込み信号を元にコイルに流す電流の向きを変え、磁化パターンを記録する
- 読み込み信号処理
 - ヘッドのコイルを流れる電流の変化から電流の変化したところを取り出して記録した信号を読み出す。





ハードウェアとしてのディスク

- **CHS (シリンダ、ヘッド、セクタ)モード**
 - 古くは[シリンダー番号、トラック番号、セクタ番号]でセクタを指定
 - 物理的セクタから論理的セクタへ変換
 - サイズは512バイトが典型的
 - ファイルシステムのブロックサイズは1～8KBが典型的
 - 不良セクタがあったとき、代替セクタを割り当てられるようにするため
 - 外周でより多くのセクタを割り当てられるようにするため
 - 高密度化
- **LBA (Logical Block Addressing) モード**
- **ディスク容量**
 - **CHSモード**
 - シリンダー数 \times ヘッド数 \times セクタ数 \times セクタサイズ
 - **LBAモード**
 - セクタ数 \times セクタサイズ





3.3 ファイル管理—ファイルシステム—

3.3.2 ファイルの管理と操作

[1] ファイルシステム

- OSのファイル管理機能
 - ファイルシステムというシステムサービスを担うOSモジュールによって実現
- ファイルシステム
 - コンピュータシステムのユーザに対して、ファイルに対する統一的なあるいは一元化した管理および操作機能を提供する
- ファイルシステムが提供する具体的な機能
 1. ファイルとファイル装置との対応付け
 - ファイルにファイル装置の領域を割り当てる機能(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とアクセス方式を統一してユーザに提供
 - ユーザに対して提供する「ファイルへの簡便なアクセス方式」はOSが実現するユーザインタフェースの一部
 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する(ファイル保護)



3.3 ファイル管理ーファイルシステムー

3.3.2 ファイルの管理と操作

[1] ファイルシステム

- OSのファイル管理機能
 - ファイルシステムというシステムサービスを担うOSモジュールによって実現
- ファイルシステム
 - コンピュータシステムのユーザに対して、ファイルに対する統一的なあるいは一元化した管理および操作機能を提供する
- ファイルシステムが提供する具体的な機能

1. ファイルとファイル装置との対応付け	<ul style="list-style-type: none">• ファイルにファイル装置の領域を割り当てる機能(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とアクセス方式を統一してユーザに提供	<ul style="list-style-type: none">• ユーザに対して提供する「ファイルへの簡便なアクセス方式」はOSが実現するユーザインタフェースの一部
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する(ファイル保護)	



3.3 ファイル管理ーファイルシステムー

3.3.2 ファイルの管理と操作

[2] ファイル操作

- ファイルシステムが提供するファイルに対する統一的なあるいは一元化した操作機能

(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」通知を受け付けて、当該ファイルに対する管理機能を開始する
(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通知を受け付けて、当該ファイルに対する管理機能を終了する
(3) 生成	<ul style="list-style-type: none">ファイル装置上でファイル格納のために領域を確保する。I確保した領域にファイルを格納する(5)の書き込み操作を含める場合もある
(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する
(5) 書き込み(ライト、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)	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)	ファイルのコピーを作る
(8) 移動	ファイル装置上でファイルの格納領域すなわち場所を変更する
(9) リスト	指定した領域に格納しであるファイルの属性(名称、サイズ、アクセス日時など)についての一覧表(リスト)を出力する



3.3 ファイル管理—ファイルシステム—

3.3.2 ファイルの管理と操作

[3] ファイル制御ブロック (FCB)

定義3.8 (ファイル制御ブロック)

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)をファイル制御ブロック(FCB)あるいはファイル記述子(ファイルディスクリプタ)という

- FCBの元に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として、ファイル装置に格納してある
 - 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(実際には、メインメモリ)上のFCB領域にコピー
 - ファイル使用中は、その仮想メモリ上のFCB領域で、FCBとして保持
 - 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、仮想メモリ上のFCB領域から書き戻しする、すなわち更新

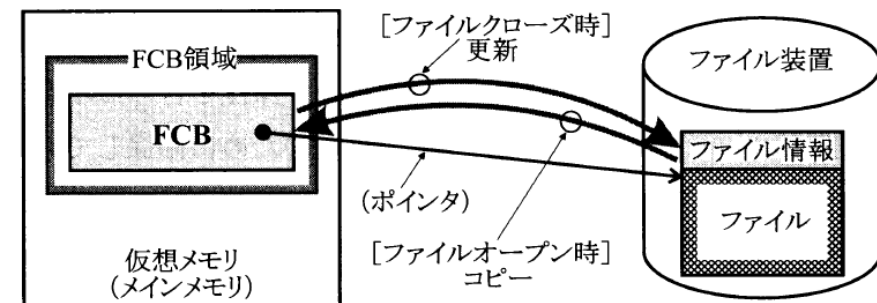


図 3.20 ファイル制御ブロック (FCB)



3.3 ファイル管理—ファイルシステム—

3.3.2 ファイルの管理と操作

[3] ファイル制御ブロック(FCB)

- ファイルへのポインタはメインメモリ上のFCB(領域)で、ファイルそのものはファイル装置上で、それぞれ保持し管理する
 - (1)~(4) や(9) のファイルそのものの管理・制御機能、(5)~(8) の『実際のファイル装置上でのファイル操作』の管理・制御機能のほとんどは、仮想メモリ上にあるFCBに対する操作となる
 - ファイル装置上のファイルへの直接のアクセスが生じるのは、(5)~(8) のファイル操作だけ

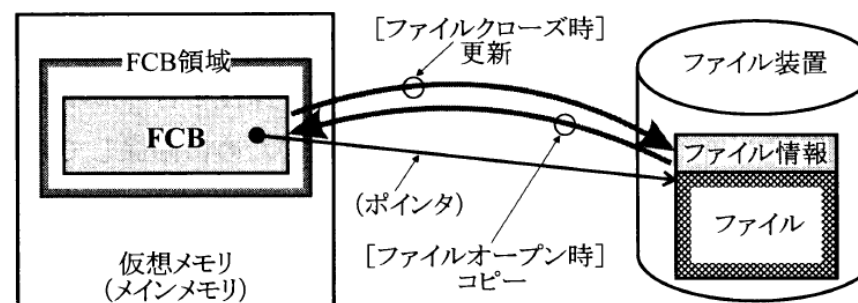


図 3.20 ファイル制御ブロック (FCB)

- | |
|------------------|
| (1) オープン(open) |
| (2) クローズ(close) |
| (3) 生成 |
| (4) 消去(削除) |
| (5) 書き込み(ライト、更新) |
| (6) 読み出し(リード) |
| (7) コピー(複写) |
| (8) 移動 |
| (9) リスト |



3.3 ファイル管理ーファイルシステムー

3.3.2 ファイルの管理と操作

[4] ファイル情報(例)

項目	内容
(1)名前	ユーザが付けた名前(論理名)、ファイル名
(2)型	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、データなどのファイル内容の種類
(3)サイズ	ファイルが占める領域の大きさであり、ファイルブロック総数で示す
(4)構造に関する情報	FAT、UTS、NTFSなどのファイル割り付け方式
(5)保護に関する情報	読み出し可能 (Readable; R)、書き込み可能 (Writable; W)、実行可能 (eXecutable; X) などのアクセス権の別、および、所有者 (owner; 作成者) や使用者 (user; ユーザ) などの当該権限を有する者の識別であり、属性ともいう
(6)参照局所性に関する情報	作成日時、(最終)更新日時、(最終)アクセス日時などの時間情報、読み出し回数や書き込み回数などの統計情報
(7)アドレス	格納するあるいは対応付けたファイル装置(実際にはボリューム)上の物理アドレスすなわちポインタ

ディレクトリ

(1)名前と(7)アドレス、および、名前とアドレスとの対応付け(ファイル割り付)の各情報→3.3.4項

ファイル割付方式
→3.3.5項

権限、ファイル保護
→3.3.6項

ボリューム→3.3.5項



3.3 ファイル管理ーファイルシステムー

3.3.2 ファイルの管理と操作

[5] ファイルのオープンとクローズ

- (1)のファイルオープンと(2)のファイルクローズのファイル操作
→プロセス(プログラム)自身が、実行中に使用するファイルについて、OSにファイル使用の開始(1)、ファイル使用の終了(2)を通知あるいは宣言する機能

(1) オープン操作の手順

1. OSが仮想メモリ上にFCB領域を確保する
2. 使用ファイル(名)を特定すなわち識別する
3. そのファイルのヘッダ部にあるファイル情報を、FCBの項目としてFCB領域に設定かつコピーする

ユーザがやるのではなく、OSがやってくれるところに意味がある

(2) クローズ操作の手順

1. FCB領域のうちの更新されている項目によって、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する
2. 仮想メモリ上のFCB領域を解放する、すなわち、OSに返却する



3.3 ファイル管理—ファイルシステム—

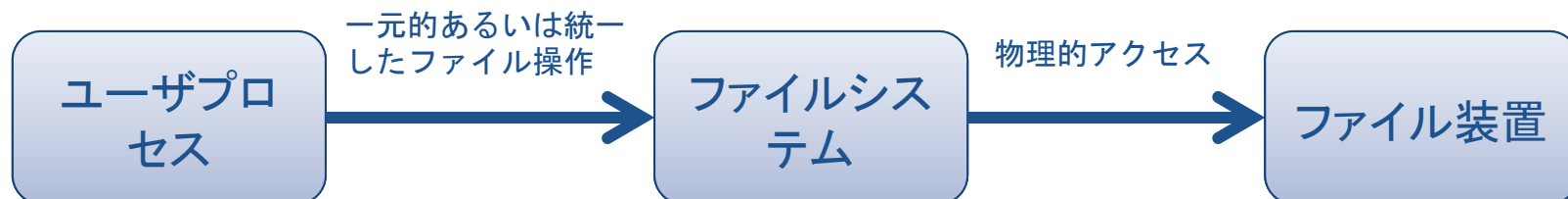
3.3.3 ファイル構造とファイルアクセス方式

[1] ファイル構造とファイルアクセス方式

定義3.9 (ファイル構造)

ファイルシステムが規定する「ファイルの論理的な構造(論理構造)」をファイル構造あるいはファイル編成という。

- 一元的あるいは統一した、ファイル構造の提供
 - 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「一元的あるいは統一したファイル操作」を実現する
 - 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、すなわち、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する





3.3 ファイル管理—ファイルシステム—

3.3.3 ファイル構造とファイルアクセス方式

[1] ファイル構造とファイルアクセス方式

定義3.10 (ファイルアクセス方式)

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(読み出しと書き込みの総称)方式をファイルアクセス方式という。ファイルアクセス方式もファイルシステムが規定する。

- 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あるいは実装する
- 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てしまうこともある。

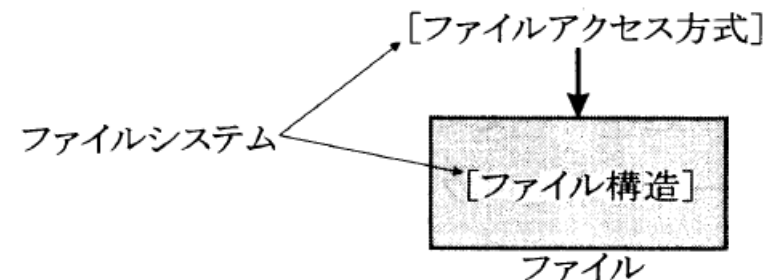


図 3.21 ファイル構造とファイルアクセス方式



3.3 ファイル管理—ファイルシステム—

3.3.3 ファイル構造とファイルアクセス方式

[1] ファイル構造とファイルアクセス方式(続き)

- ファイル構造を含むファイルアクセス方式の目的: ファイルそのものおよびファイルへのアクセスの透明化
 - 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をまったく意識することなく、ファイル構造にアクセスできること
 - 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あるいは統一によってファイルアクセスの透明化を実現する
- ファイルアクセス方式の種類
 - (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)アクセスする
 - (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すなわち順不同でアクセスできる。ランダムアクセス(任意順アクセス)
- ファイル装置に対するアクセス方向によって
 - (a) 読み出し(ロード): ファイル装置からメインメモリへファイルを転送する操作
 - (b) 書き込み(ストア): メインメモリからファイル装置へファイルを転送する操作

参考3.12 (透明化)

ソフトウェアで実現する論理的機能によって、別の機能を隠ぺいすること。特に、ハードウェア機構で実現している物理的機能を隠ぺいする場合に使うことが多い。



3.3 ファイル管理—ファイルシステム—

3.3.3 ファイル構造とファイルアクセス方式

[2] 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でファイルブロックを並べるファイル構造(論理構造)
- ファイルアクセス方式は、逐次アクセスが普通
- アクセス順はファイルブロックの並び順そのままに変更できない。すなわち、固定
- 逐次アクセスファイル装置
 - 磁気テープ装置
 - ファイルブロックをファイル装置上の連続する領域に格納
 - ファイルに逐次的にアクセスする
 - ファイル構造は逐次アクセスファイルに限るのが普通
- 長所
 - 連続ブロックの逐次アクセス(読み出しと書き込み)は高速
 - 読み出しと書き込みに限ると、ファイル操作は単純
- 短所
 - 読み出しと書き込み以外の、生成(追加)や(一部分の)削除などのファイル操作は面倒
 - ファイル途中でのブロックの挿入や削除が困難

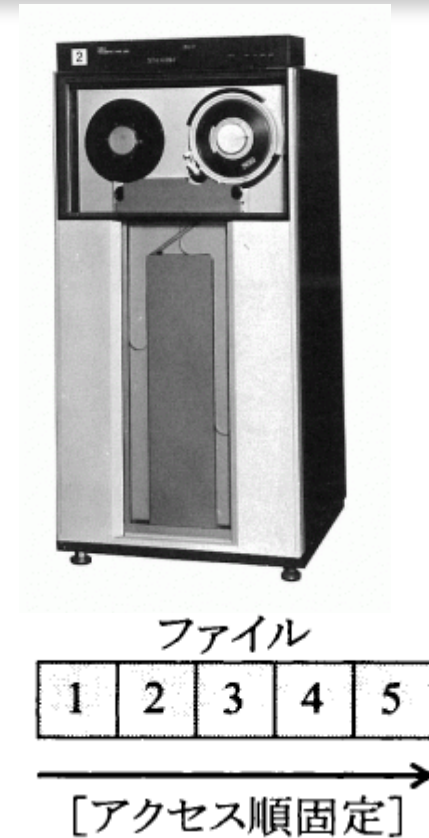


図 3.22 逐次アクセスファイル



3.3 ファイル管理—ファイルシステム—

3.3.3 ファイル構造とファイルアクセス方式

[2] ファイル構造—代表例—(続き)

(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
- ブロック番号(相対ブロック番号)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)できる。ランダムアクセスファイルともいう
- ハードディスクドライブ装置
 - ファイル装置上のどのファイルブロックにも直接アクセスできる
 - 直接アクセスファイル装置あるいはランダムアクセスファイル装置
 - 「ドライブ」が付くファイル装置(光ディスクドライブ装置など)やフラッシュメモリは直接アクセスファイル装置
 - 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

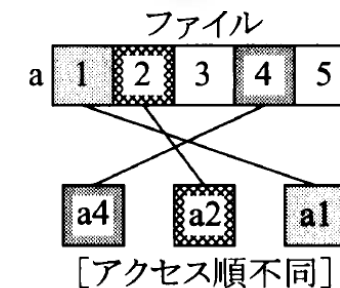


図 3.23 直接アクセスファイル

定義3.11 (相対ブロック番号)

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)』を相対ブロック番号という。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する



3.3 ファイル管理ーファイルシステムー

3.3.3 ファイル構造とファイルアクセス方式

[2] ファイル構造ー代表例ー

(B) 直接 アクセスファイル(続き)

- 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できる
→特定ブロックへのアクセスが高速
 - ファイル装置への格納方法はOSのファイルシステムごとの自由な設定が可能
- 短所
 - OSのファイルシステムごとの管理が必須
 - 管理が複雑



3.3 ファイル管理—ファイルシステム—

3.3.3 ファイル構造とファイルアクセス方式

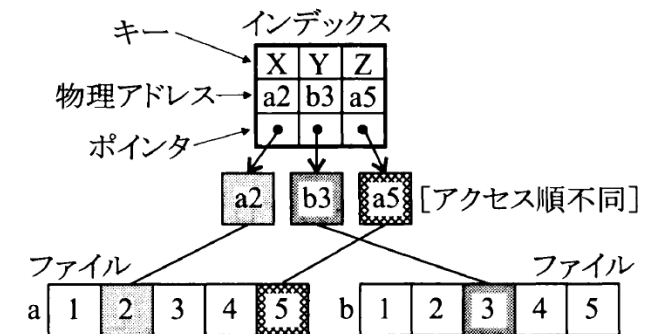
[2] ファイル構造—代表例—(続き)

(C) インデックス付きファイル

(1) キーによってインデックスを検索(連想)

(2) そのキーと対になっている物理アドレスによって、当該ブロックに直接アクセスの順でアクセス

- 直接アクセスファイル装置とのマッピングが容易であり、親和性が高い
- 長所
 - 論理的な逐次アクセスも、キーによる直接アクセスもどちらも可能
- 短所
 - インデックス管理やインデックス検索のコストが大きい
 - ディスクにインデックスをおくと、ディスクに2回アクセスしないといけない



定義3.12(インデックス)

配列や表など均質に並んでいる項目のうち、特定の項目を指す数字の指標、または、それを参照するためのキー(文字列が普通)をインデックス(索引)という。ファイルアクセス方式においては、ブロックの『論理名』を示すキーとファイル装置上の物理アドレスの対をインデックスとしている。



3.3 ファイル管理ーファイルシステムー

3.3.4 ディレクトリ管理

[1] ディレクトリ

定義3.13(ディレクトリ)

ファイルシステムにおいてファイル(作)に関する情報」すなわちファイルで重要な項目だけを抜粋して、か一覧表)をディレクトリという

- ファイル情報(項目)のうちの

- (a) ファイル名(論理名)

- (b) そのファイルが格納してある場所、すなわち、ファイル表直上(実際にはファイル表直上の1個のメモリ領域区分)上の物理アドレス

- (c) (a)のファイル名と(b)のファイルがある場所との対応付け(ファイル割り付け)に関する情報

- OS(ファイルシステム)自身が、ファイル管理およびファイル操作のためにディレクトリを参照あるいは更新する

- OSは、ディレクトリを保持して、その情報をユーザプログラムに提示あるいは提供する

項目	内容
(1)名前	ユーザが付けた名前(論理名)、ファイル名
(2)型	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、データなどのファイル内容の種類
(3)サイズ	ファイルが占める領域の大きさであり、ファイルブロック総数で示す
(4)構造に関する情報	FAT、UTS、NTFSなどのファイル割り付け方式
(5)保護に関する情報	読み出し可能 (Readable; R)、書き込み可能 (Writable; W)、実行可能 (eXecutable; X) などのアクセス権の別、および、所有者 (owner; 作成者) や使用者 (user; ユーザ) などの当該権限を有する者の識別であり、属性ともいう
(6)参照局所性に関する情報	作成日時、(最終)更新日時、(最終)アクセス日時などの時間情報、読み出し回数や書き込み回数などの統計情報
(7)アドレス	格納するあるいは対応付けたファイル装置(実際にはボリューム)上の物理アドレスすなわちポインタ



3.3 ファイル管理ーファイルシステムー

3.3.4 ディレクトリ管理

[1] ディレクトリ(続き)

- ディレクトリ管理

- 実際の「ディレクトリへの登録」
 - (a)~(c) の各情報を確定
 - (a)のファイル名と(b)のディレクトリ名
- ディレクトリ上でのファイルの格納
特に、ボリュームのディレクトリ

- ディレクトリそのものもファイル

- ファイルアクセスは、「(1)ディレクトリへのアクセス」から成る
 1. ファイルがあるファイル装置上の物理アドレスを得て
 2. その物理アドレスによって、ファイルそのものへのアクセスを行う

- ファイルシステムにおけるディレクトリの利用例

- ファイル検索: ファイル名(論理名)によってファイル装置上でのファイルの格納場所(物理アドレス)を検索する
- ファイル保護: ファイル操作時に、当該ファイル操作に関する権限をチェックする→

3.3.6項

項目	内容
(1)名前	ユーザが付けた名前(論理名)、ファイル名
(2)型	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、データなどのファイル内容の種類
(3)サイズ	ファイルが占める領域の大きさであり、ファイルブロック総数で示す
(4)構造に関する情報	FAT、UTS、NTFSなどのファイル割り付け方式
(5)保護に関する情報	読み出し可能 (Readable; R)、書き込み可能 (Writable; W)、実行可能 (eXecutable; X) などのアクセス権の別、および、所有者 (owner; 作成者) や使用者 (user; ユーザ) などの当該権限を有する者の識別であり、属性ともいう
(6)参照局所性に関する情報	作成日時、(最終)更新日時、(最終)アクセス日時などの時間情報、読み出し回数や書き込み回数などの統計情報
(7)アドレス	格納するあるいは対応付けたファイル装置(実際にはボリューム)上の物理アドレスすなわちポインタ



3.3 ファイル管理ーファイルシステムー

3.3.4 ディレクトリ管理

[2] ディレクトリ管理

- ディレクトリ管理は以下を併せた統一的な機能として実現する
 - OSによるファイル管理
 - プログラム(OSやユーザ)によるファイル操作
- ディレクトリ管理の具体的な機能
 1. 検索
 - ファイル名(論理名)でディレクトリを検索し、ファイルのファイル装置上でのアドレス(物理アドレス)を得る
 2. 事項の追加と削除
 - ファイルごとにディレクトリ(の内容)を更新する
 3. リスト
 - ファイル情報を一覧表示する
 - ディレクトリそのものがリスト(一覧表)であるので、ディレクトリをそのまま、あるいは、要求によっては並び替えて、出力する



3.3 ファイル管理—ファイルシステム—

3.3.4 ディレクトリ管理

[3] ディレクトリの構造—代表例:木構造ディレクトリー

- ディレクトリの構造＝ディレクトリの構成方式
- 木構造ディレクトリの構成
 - ディレクトリ全体を1個の木構造とする
 - その木の唯一ルート(根)とすべてのノード(節)をディレクトリとする。その唯一ルートがルートディレクトリ
 - 各ディレクトリは、それが管理する部分木のルートディレクトリ
 - あるディレクトリ、すなわち、その部分木のルートに属するファイルは、その部分木の葉として置く
- 木構造ディレクトリによるファイルへのアクセスは論理的なパス(経路)名で行う

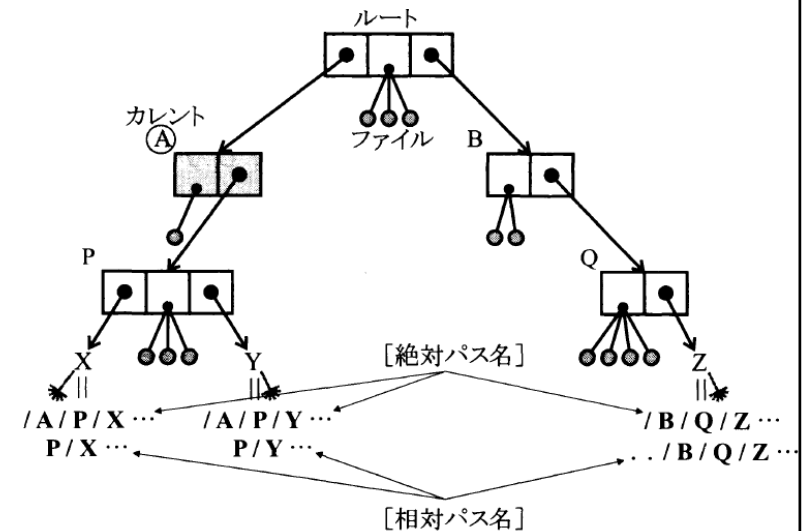


図 3.25 木構造ディレクトリ (例)

定義3.14 (パス名)

あるディレクトリ(始点、木構造でのルートかノード)からアクセス対象(終点)であるファイル(木構造での葉)や別のディレクトリ(木構造でのノード)に至る論理的なパス上にあるディレクトリ名を、あらかじめ規定してある規則にしたがって、順次連結して作る文字列をパス名という



3.3 ファイル管理—ファイルシステム—

3.3.4 ディレクトリ管理

[3] ディレクトリの構造—代表例:木構造ディレクトリー (続き)

- 木構造ディレクトリにおける「パス名の付け方」には、始点のディレクトリがルートであるか、ノードであるかによって、次の2種類に大別できる
 - 絶対パス名
 - 木全体の唯一ルートであるルートディレクトリを始点のディレクトリとする
 - そのルートディレクトリから対象ファイルに至るパス(絶対パスあるいは完全パス)上のディレクトリ名を、“/”で区切って順次連結して作る文字列
 - どのファイルに対しても、ルートディレクトリからの絶対パス名は唯一である
 - 相対パス名
 - カレントディレクトリを始点のディレクトリとする
 - カレントディレクトリから対象ファイルに至るパス(相対パス)上のディレクトリ名を“/”で区切って順次連結して作る文字列
 - あるファイルへの相対パスは、カレントディレクトリによって決まり、相異なる

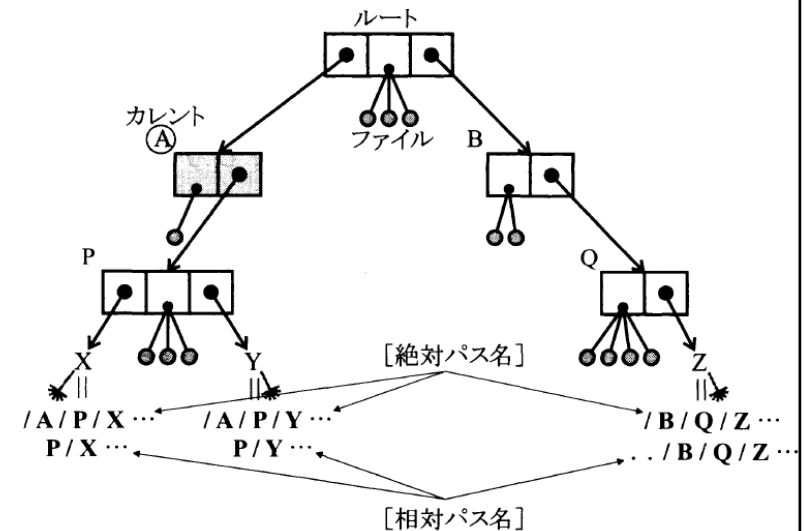


図 3.25 木構造ディレクトリ (例)



参考3.13(木構造)

- 木構造
 - 木形状のリストであるデータ構造、あるいは木形状のリストに構造化してあるデータ
 - ルート: リストの始端データをルート
 - ノード: 途中データ
 - 葉: 末端データ
 - 一般的なリストを『ルート(根)からノード(節)を経て葉へ展開する』、あるいは逆に『葉からノード(節)を経てルート(根)へ縮約する』という「構造化」によって簡潔にしたリスト
 - 木構造の一部分(のリスト)を部分木であり、木は「部分木の集まりで構成し、逆に、部分木の集まりに分解できる

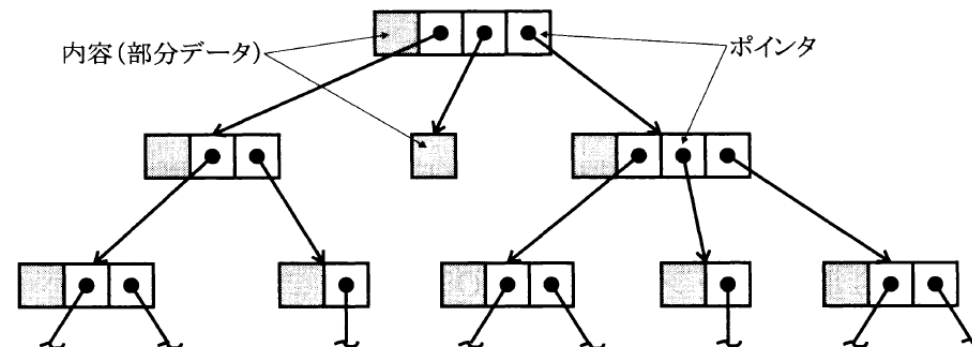


図 3.26 木構造 (参考 3.13)