

受講にあたっての注意事項

本講義の全部あるいは一部の
録画・録音および複製ならび
に再配布を、厳に**禁じます**。



大阪大学 基礎工学部

SCHOOL OF ENGINEERING SCIENCE



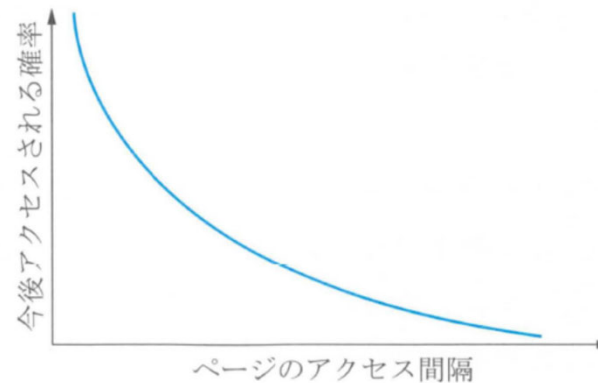
第11回(5月21日1時限)ミニレポート課題

1. LRUアルゴリズムの動作例を、参照ストリングを“0 1 2 3 4 5 1 0 3 2 0 4 5 1”、ページ枠を4とした場合について、LRUスタックの変化の様子を示せ。また、実メモリ上のページ枠の内容の変化の様子を別に示せ。
2. 次にアクセスされる確率をもっとも低いページを選択する近似的な手法としてLRUアルゴリズムが用いられる理由を説明せよ。
3. LRUアルゴリズムを正確に実装することが困難な理由を説明せよ。



第11回(5月21日1時限)ミニレポート課題 解答

1. 省略
2. ページが今後アクセスされる確率と、そのページの過去のアクセス間隔との間には図に示すような関係があることが経験的に知られている。



3. LRUアルゴリズムは、CPUが主記憶にアクセスするたびに、①時刻を調べて、②テーブル内のページアクセス時刻を更新する必要がある。主記憶へのアクセスはほぼCPUの1マシンサイクルごとに発生し、そのたびごとにページアクセス時刻を更新するオーバーヘッドが膨大になるため。また、ページフォールトが起こった場合は、ページテーブルのアクセス時刻からもっとも古くにアクセスされたページを探し出す必要がある。



第12回(5月21日2時限)ミニレポート課題

1. 以下のページング方式におけるページ読込のタイミングに関する説明について(a)~(h)の空白を埋めよ。

ページング方式においては、どのタイミングでページを仮想アドレス空間から実アドレス空間に読み込むかが重要であり、大別して(a **デマンドページング**)と(b **プリページング**)がある。

(a **デマンドページング**)では、プログラム自身が実行時にアクセスや参照を要求するページを読み込む。すなわち、OS機能を(c **ハードウェア機構**)が支援する(d **動的**)ページングである。長所としては、必要なページだけを読み込むため無駄がないことが挙げられる。一方、短所としては、プロセス実行の(e **開始時**)にページフォールトが集中発生することが挙げられる。

一方、(b **プリページング**)は、アクセスや参照があると予測されるページをOSが実行前にまとめて読み込むものであり、OS機能を(f **コンパイラ**)が支援する(g **静的**)ページングである。

現代では、(h **メインメモリ**)の実装コストが低くなったので、余分で冗長なページを読み込んでおいても予測がはずれた場合の影響は少ないため、(b **プリページング**)を併用するOSが多い。

2. 上記(b **プリページング**)の方式の長所、短所を挙げよ。

長所

(1) 予測的中率が高ければ、複数ページを一度に読み込むブロック転送が適用できるので、読み出し回数とページ当たりの平均読み出し時間は減り、高速化が達成できる;

短所

(1) 不要なページも読み込んでしまう可能性がある

(2) 予測的中率が低いと、予測に要するオーバーヘッドが顕在化する。予測に要するオーバーヘッドを上回る(ページ置換そのものの)高速化が必須の要件となる

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 3.3.4 ディレクトリ管理



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—

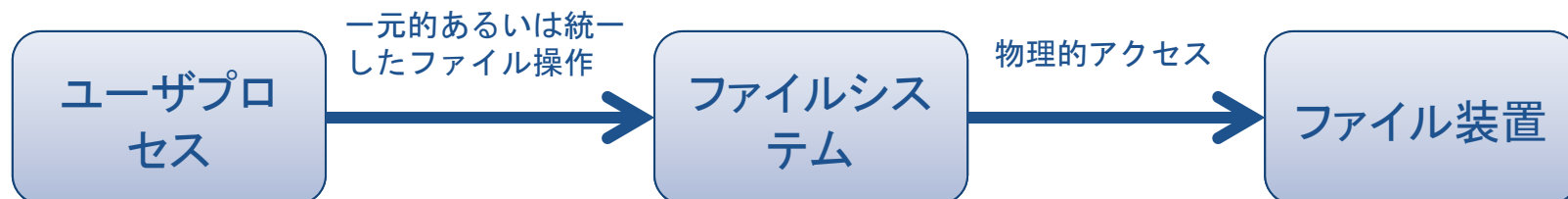
3.3.3 ファイル構造とファイルアクセス方式

[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。

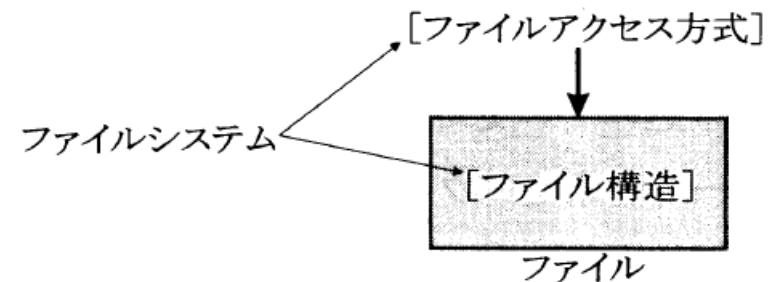


図 3.21 ファイル構造とファイルアクセス方式



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(プログラマ)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
- 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)

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

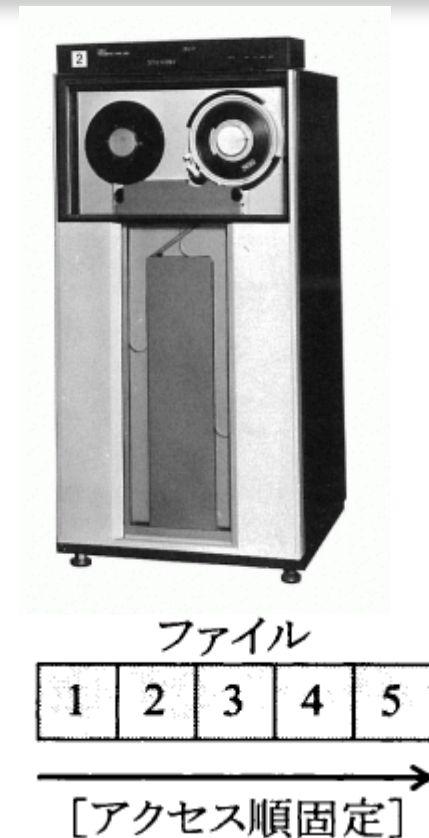


図 3.22 逐次アクセスファイル



古くて新しい磁気テープ

<https://it-trend.jp/data-backup/article/currentevent>

 IT製品の比較・資料請求サイト



資料請求リスト 0件 [資料請求フォームへ](#)

[ホーム](#) [カテゴリから探す](#) [ランキングから探す](#) [記事を読む](#) [セミナーを探す](#) [はじめての方へ](#)

IT製品比較TOP > データ管理 > データバックアップ製品 > データバックアップ製品の関連記事一覧 > 古くて新しい磁気テープ、まさかの復権？データバックアップ利用におけるメリットとは？

古くて新しい磁気テープ、まさかの復権？データバックアップ利用におけるメリットは？

Share

Tweet

B! Hatena

Pocket

2019年11月28日 最終更新

データバックアップ製品の製品一覧



企業の大切なデータをバックアップするストレージとして、入出力速度が高くリストア（復旧）も含めて、短時間で処理できるハードディスクドライブ（HDD）の活用が主流とされていますが、最近になって、アナログなメディアながらも磁気テープが再び注目され始めています。

ここでは磁気テープ復権の理由を詳細に探ることにより、企業のデータバックアップのあり方を効率的に改善していく方法を解説していきます。

データバックアップ製品の製品を調べて比較



製品をまとめて資料請求！ [資料請求フォームはこちら](#)

データバックアップ製品の資料請求ランキングで製品を比較！ 今週のランキング第1位は？

目次 [\[非表示\]](#)



資料請求ランキング

7月13日(月) 更新

いざという時の備え「今」始めませんか？BCPリモートバックアップサービス株式会社 伊藤 康平

データセンターでしっかりバックアップ！RTO・RPOバックアップサービス株式会社 北原 慎一

迅速なバックアップと即座のリカバリでRTOとRPOに対応！Rapid Recovery クエストソフトウェア株式会社

[一覧を見る](#)

[ログイン](#)

[新規会員登録](#)

ITトレンドに貴社製品も掲載しませんか？
[詳しくはこちら](#)

新着記事

中小企業にデータバックアップは必要？企業が注意すべきことを解説

中小企業でもデータバックアップは必要なのでは？

無料・無料別データバックアップソフト13選！どれを選ぶべきか解説

■LTO(Liner Tape Open)の大容量化

形式	発売	容量		転送速度	
		非圧縮時	圧縮時	非圧縮時	圧縮時
LTO-6	2012年	2.50TB	6.25TB	160MB/秒	400MB/秒
LTO-7	2015年	6TB	15TB	300MB/秒	750MB/秒
LTO-8	—	12.8TB	32TB	472MB/秒	1180MB/秒

電力コスト削減

移動時の携帯性

価格優位性

長期保管性





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

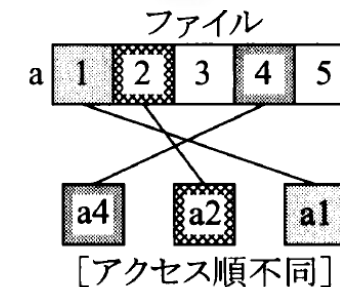


図 3.23 直接アクセスファイル

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

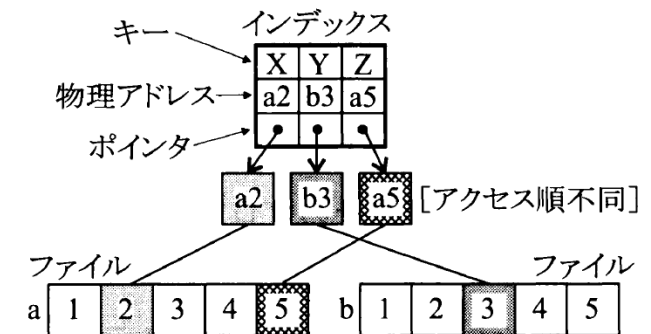


図 3.24 インデックス付きファイル

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—ファイルシステム—

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装置に格納してある「すべてのファイル(の管理や操作)に関する情報」すなわちファイル情報のうちから、**ファイル管理とファイル操作に必須で重要な項目だけ**を抜粋して、かつ、それらを体系化および一元化して構成するリスト(一覧表)をディレクトリという

- ファイル情報(項目)のうちの以下で構成する
 - (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、ファイル装置(実際にはファイル装置上の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



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」は
 - (a)~(c) の各情報を確定して
 - (a)のファイル名と(b)のボリュームとの対応付けである(c)のファイル割り付けを行うこと
 - ディレクトリ上でのファイル名とボリュームとの対応付け(マッピング)操作機能のうち、特に、ボリュームのディレクトリへの登録操作をマウントという
- ディレクトリそのものもファイル装置上に置く
 - ファイルアクセスは、(1)ディレクトリ、(2)ファイルそのもの への2回の「ファイル装置へのアクセス」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



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

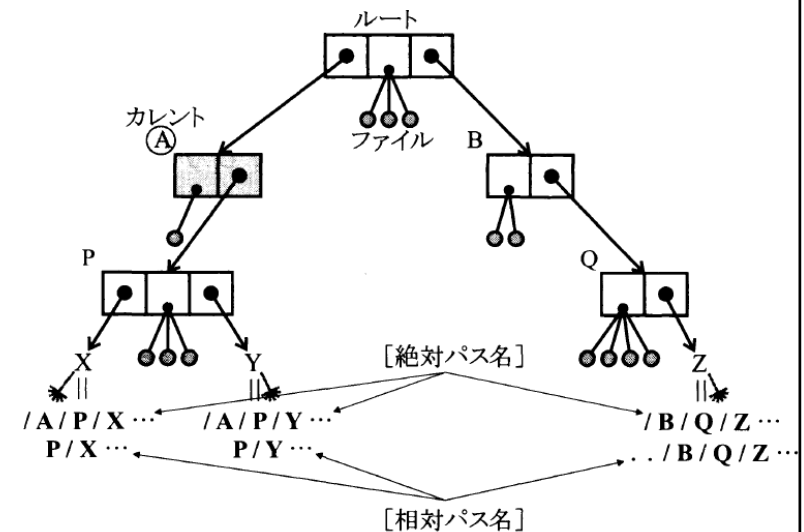


図 3.25 木構造ディレクトリ (例)

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

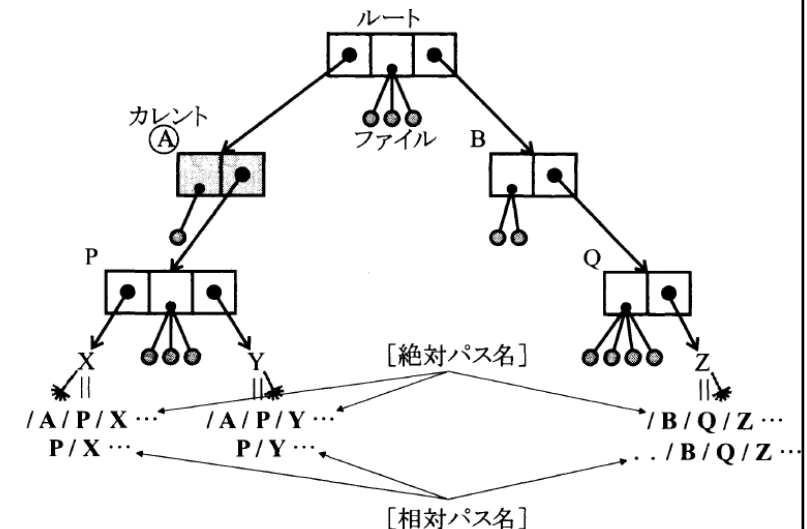


図 3.25 木構造ディレクトリ (例)



第13回(5月28日1時限)ミニレポート 課題

カッコ内を埋めよ。

ファイルへのアクセス方式は大きく分けて、(①)方式と(②)方式がある。(①)方式は主に(③)装置で用いられ、(②)方式は(④)装置で用いられる。

(④)装置は、(⑤)番号、(⑥)番号、(⑦)番号を指定することにより、ディスクのどの位置でも読み書き可能であり、最少のアクセス単位は(⑦)となる。

ユーザにとって、自由でよりわかりやすい名前をつけることができるために開発されたファイルシステムが(⑧)である。(⑧)には、現在プログラムが注目しているディレクトリ上の位置を示す(⑨)とよばれる概念が存在し、(⑨)からみた注目ファイルの名前表現を(⑩)とよび、ルートを起点とする名前表現を(⑪)とよぶ。

締切:6月2日(水)
CLEで提出



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)は部分木と呼び、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

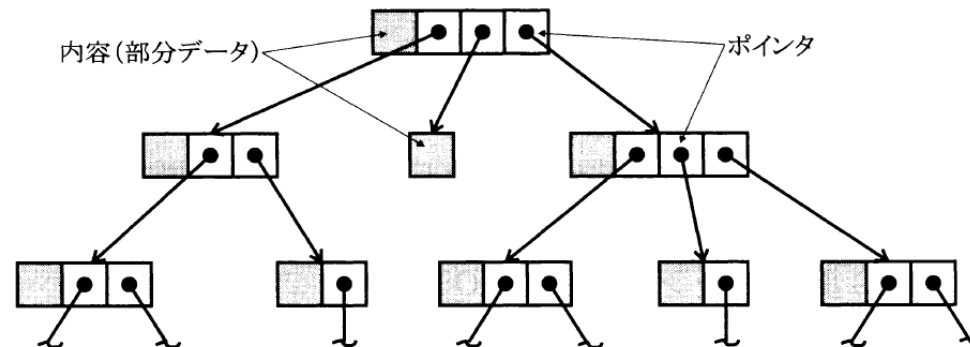


図 3.26 木構造 (参考 3.13)