

ファイルシステムの 機能3 アクセス方式

ファイルの概念とその操作

- ファイル=データを格納（保存）する入れ物
- 内容： いろいろな場合がある
 - バイトの連なりの場合
 - 文字・数値の羅列 ⇒ 解釈は読み出したソフト次第
 - 構造をもつ場合 ⇒ 例：レコードを単位にした情報
 - レコードごとの読出しをOSがサポートする場合
 - データベースシステムがサポートする場合

レコード→

氏名	番号	入学年度	生年月日	住所	電話番号
氏名	番号	入学年度	生年月日	住所	電話番号
氏名	番号	入学年度	生年月日	住所	電話番号
氏名	番号	入学年度	生年月日	住所	電話番号

ファイル操作の抽象化

- 基本操作： open, read, write, close
 - 媒体等に依らず共通化 ~ 抽象化する効果あり
- read, write (読み、書き)
- open
 - ファイルの利用宣言
 - 名前（ユーザが指定）から実体への関連付け
 - 特に複数の同時アクセス時に、資源排他アクセス管理
 - 「アクセス方式」（後述）の指定
 - など
- close openの逆（終り）



アクセス方式の考え方

- アクセス方式 = 読み方（アクセスの仕方）の違い



アクセス方式の考え方

- アクセス方式 = 読み方（アクセスの仕方）の違い
- 順（シーケンシャル）アクセス方式
- 直接（ランダム）アクセス方式

4

アクセス方式の考え方

- アクセス方式 = 読み方（アクセスの仕方）の違い
- 順（シーケンシャル）アクセス方式
 - 先頭から順番に読む（ユーザは場所指定しない）
 - イメージとしてテープ（ビデオや音楽カセット）
- 直接（ランダム）アクセス方式

5

アクセス方式の考え方

- アクセス方式 = 読み方（アクセスの仕方）の違い
- 順（シーケンシャル）アクセス方式
 - 先頭から順番に読む（ユーザは場所指定しない）
 - イメージとしてテープ（ビデオや音楽カセット）
- 直接（ランダム）アクセス方式
 - ブロック番号を指定して、どのブロックへもアクセス
 - イメージとしてハードディスクやメモリ

6

ハードウェアの機能制約とアクセス方式

- 注意：ハード構造⇒アクセス方式ではない
- テープの場合は、ハード構造が順アクセス
 - といって、直接アクセスができないわけではない
 - 毎回巻き戻して、先頭から空読みして、指定されたブロックへ行き着いて、そこを読む
 - 非常に効率が悪いのは自明 ⇒ しないほうがいいが出来ないわけではない
- 逆にディスクは、ハード構造が直接アクセス
 - 当然、直接アクセスは効率よく行える
 - 順アクセスも、効率よく行える（番号順に読めばよい）
- ハード構造によって性能が制約されるだけ

7

(脱線) ファイル編成の考え方

- ファイル編成 = ファイル内のレコード格納法の違い
- 順編成ファイル(SAM)
 - 先頭から順番に連続してレコードを格納する
 - 順次アクセスする
 - 既存レコードの更新・追加・削除はできない
 - そこから後を全部書き直す(上書きする)必要がある
 - ファイル末尾への追加は可能
- 直接(ランダム)編成ファイル(DAM)
 - レコードのキーからブロック番号を計算してアクセス
 - ディスクなどの直接記憶媒体に向く



8

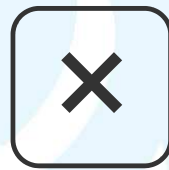
(脱線) ファイル編成の考え方

- その他、いろいろなファイル編成法が使われた(特にメインフレームOSでいろいろ用意された)
 - 索引付きファイル編成、特に索引付き順編成(ISAM)
 - 区分ファイル編成(PAM)
 - 仮想記憶編成(VSAM)
 - 自分で調べてみよ
- Windows・UNIX/Linux vs メインフレームOS
 - WinやLinuxはバイト列のみ、レコードの概念なし
 - データベースでレコード提供
 - メインフレームOSはOSとしてレコードの概念提供



9

アクセス方式の考え方が
理解できましたか？



↓
次へ

