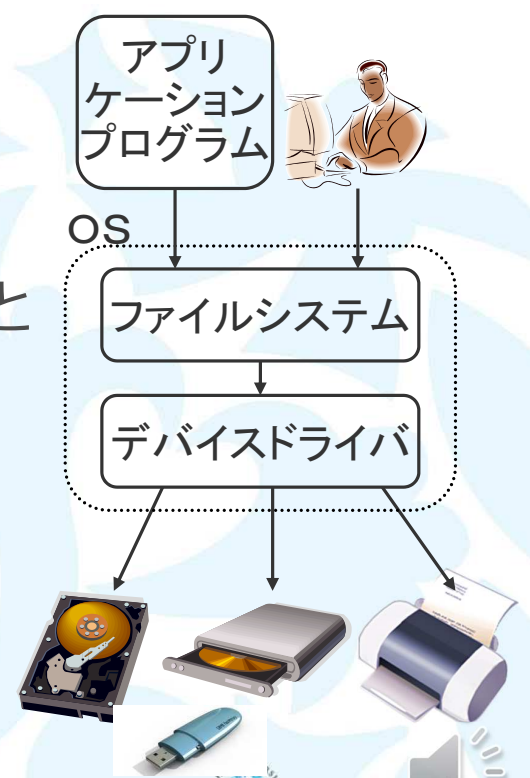


# ファイルシステムの 機能 2 データの操作

## ファイルシステムの位置付け

- ファイルシステムは、  
「ファイル」を実現する機構
- OSの一部で、
  - 利用者 (コマンドやアプリ) と  
記憶媒体の間を取り持つ
- 役割は、大別すると
  1. ファイル操作の一元化
  2. ファイルの名前付け
  3. 記憶領域の管理



# 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

2

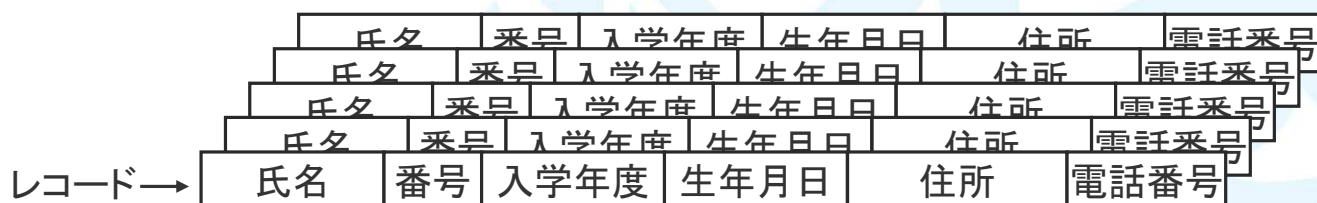
# ファイルの概念とその操作

- 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（構造が無い）
    - 文字・数値の羅列 ⇒ 解釈は読み出したソフト次第

3

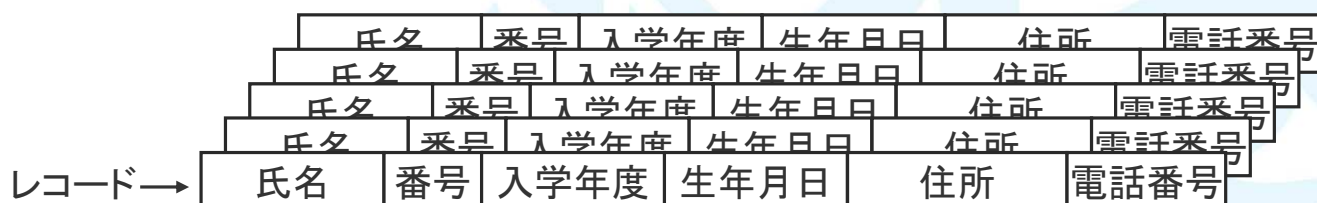
# ファイルの概念とその操作

- 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（構造が無い）
    - 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



# 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



# ファイル操作の抽象化

6



# 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

7



# 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  
ハードディスク上のファイルも  
USBメモリ上のファイルも  
プリンタ出力やキーボード入力だって  
なるべく同じ（共通の）やり方で操作

8

# 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）

9

# 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

10

# 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

11



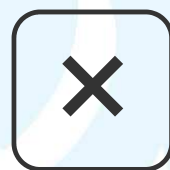
# ファイル操作の抽象化

- 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）

12



ファイルシステムの提供する  
ファイル操作の考え方が  
理解できましたか？



↓  
次へ

13

