

コンピュータ・サイエンス1

第3回

コンピュータの構成(続き), コンピュータでの情報の扱い方

人間科学科コミュニケーション専攻

白銀 純子

第3回の内容

- コンピュータの構成(続き)

前回の出席問題の解答

- 設問1: **IBM**社が発表した**PC**の仕組みと同様の仕組みで、他社が発表した**PC**の形態を何と呼ぶか?

解答: **PC/AT**互換機

- 設問2: **HDD**と**SSD**の違いについて、速度と壊れにくさの面から説明しなさい。

解答:

SSDの方がランダムアクセスの性能が良いのでアクセスが高速で、振動にも耐性があるので壊れにくい

前回の質問の回答

USBはPC用でSDカードはデジカメ用？

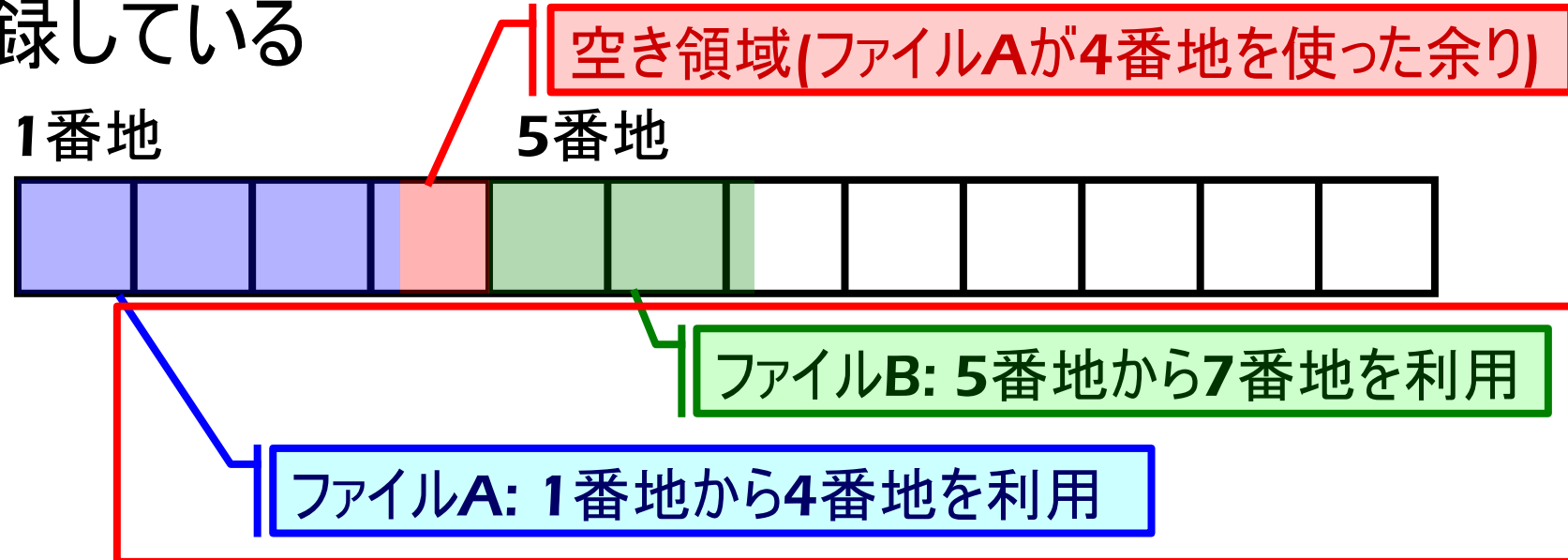
- **USB:** コンピュータへの様々なハードウェアの接続口
 - プリンタ, キーボード, マウス, **etc.**
 - 「**USBメモリ(USBフラッシュメモリ)**」は「**USB**」という接続口にさして使うための記憶装置(「フラッシュメモリ」と呼ばれる記憶装置の一種)
- **SDカード:** デジカメや携帯電話などで使われる外付けの記憶装置
 - **USBメモリ**と同じく「フラッシュメモリ」と呼ばれる記憶装置の一種(材料が同じ)
- **USBメモリ**は主にコンピュータ、**SDカード**は主にデジカメや携帯電話で利用されるが、**SDカード**を使える**PC**も存在

USBメモリ: <https://ja.wikipedia.org/wiki/USB%E3%83%95%E3%83%A9%E3%83%83%E3%82%B7%E3%83%A5%E3%83%89%E3%83%A9%E3%82%A4%E3%83%96>

SDカード: <https://ja.wikipedia.org/wiki/SD%E3%83%A1%E3%83%A2%E3%83%AA%E3%83%BC%E3%82%AB%E3%83%BC%E3%83%89>

ランダムアクセス

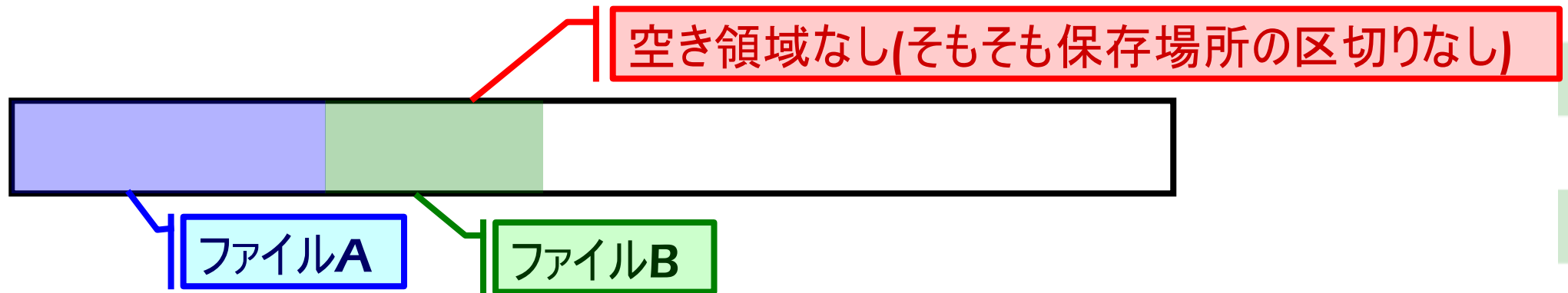
- 記憶媒体は、保存領域が一定間隔で区切られている
 - HDD**での保存領域の区切り: セクタ
 - 各セクタに番地がつけられている
 - フラッシュメモリ(**SSD**や**USBメモリ**など)での保存領域の区切り: ブロック
- 「インデックス」と呼ぶ対応関係表に、どのファイルがどの領域に保存されているかを記録している



インデックスに記録されている内容

シーケンシャルアクセス

- 記憶媒体の保存領域は区切られていない
- 空き領域なく、あるデータを保存したすぐ後に別のデータを保存する
 - 保存したデータを消して別のデータを保存するときは、前のデータへの上書きになるので、前のデータの一部が残ったり、次のデータの一部を消してしまったり



ランダムアクセスとシーケンシャルアクセス

- ランダムアクセスの記憶媒体は、インデックスをもとに、目的のセクタ・ブロックにたどる

- 他の場所に保存されているデータをチェックしない
= データの保存順序に関係なく探せる

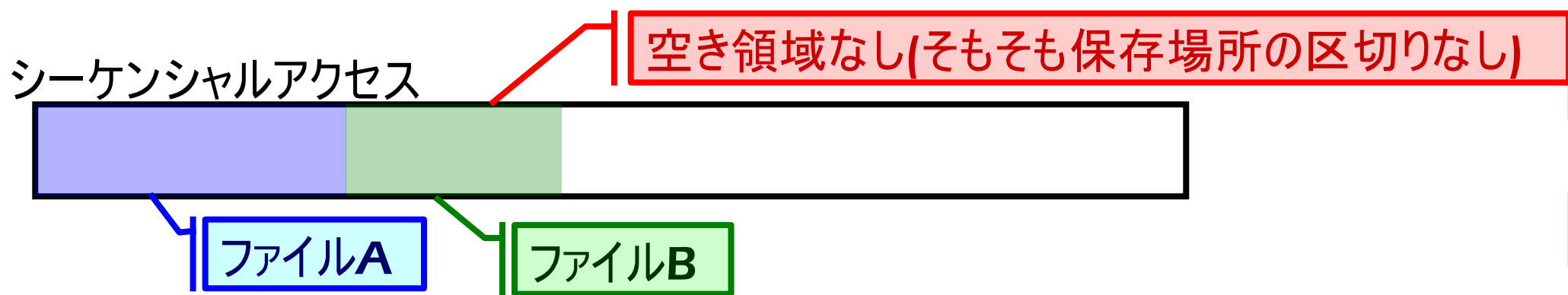
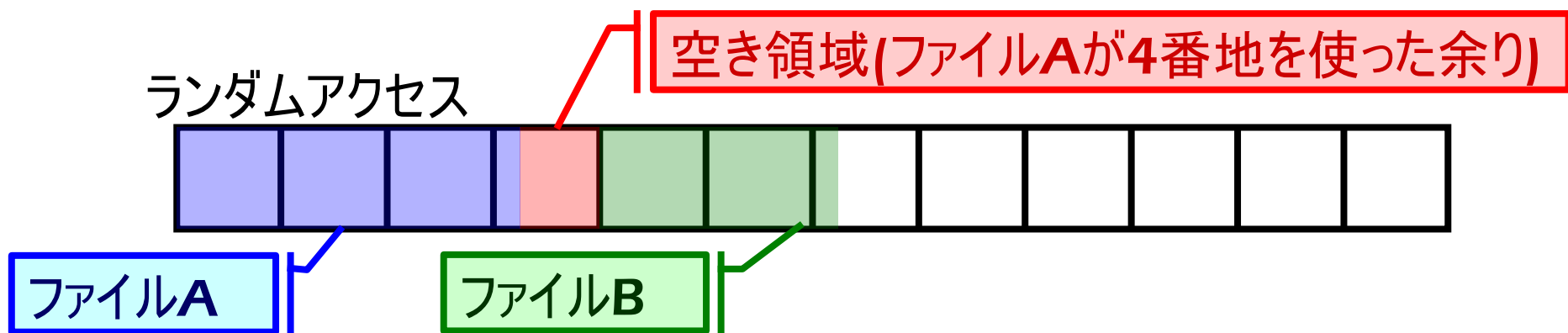
インデックスのイメージ

ファイル	保存しているセクタ・ブロック
ファイルA	1～4
ファイルB	5～7
ファイルC	8～9, 13～20

- シーケンシャルアクセスの記憶媒体は、インデックスがない
 - 先頭から順に、どんなファイルが保存されているかをチェックしていった、目的のファイルを探す
= 前の方にあるファイルにはアクセスが速く、後の方にあるファイルには遅い

シーケンシャルアクセスの利点

- あまり利点を言われることはないけど...あえて言えば:
空き領域が存在しない = 記憶媒体で無駄になる部分がない



なぜHDDがランダムアクセス?

- セクタの余りの部分にデータを保存しようとする...

- インデックスに記録する番地の情報が細くなる
 - **Ex.** ファイルCは**4.56789**番地から**4.987**番地を利用
 - = 番地を小数で正確に表現できないことも
 - = 保存されたファイルを正確に検索することが難しい
- コンピュータは、分数や無限小数は扱えない
 - **Ex.** ファイルDが8番地を3分の1だけ使うと...8番地の**8.33333...**番地以降が空き領域(**.3333...**の部分が無限に続く)
 - = インデックスに、ファイルの番地の情報を正確に記録できない

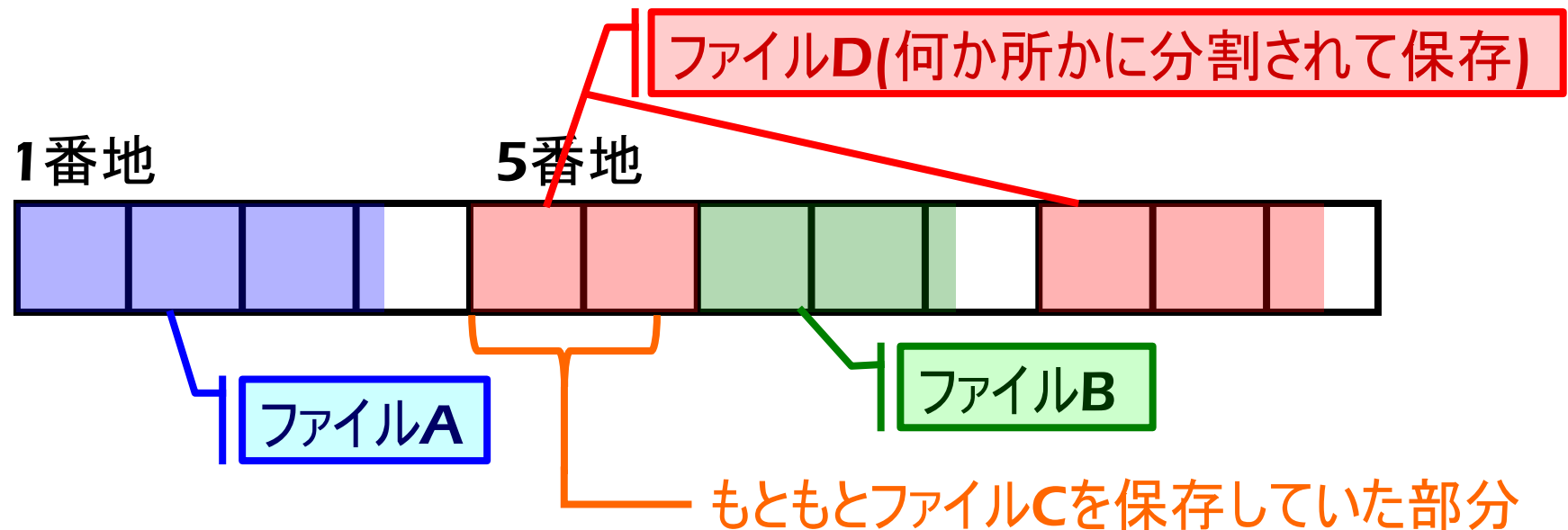
セクタの余りの部分には、他のデータを保存しない

ランダムアクセスの記憶媒体とは
インデックスを使って、ファイルの保存場所とファイルとの対応関係を管理している記憶媒体

※SSDやUSBメモリなどの円盤でない記憶媒体も、セクタではないが区切りがある

ちなみに...[1]

- 様々なファイルを保存・削除していくと...
 - 削除したファイルが保存されていたセクタが空になる
 - 1つのファイルが分割されて保存される
= そのファイルを操作するときに、時間がかかる



コンピュータを1年、2年と使い続けていると、遅くなってくる原因の1つ

ちなみに...[2]

- コンピュータが遅くなってきたと思ったら...
 - ディスクデフラグの処理をすると、断片化されたファイルをできるだけ1つにまとめて**HDD**を整理
 - 主に**Windows**で必要な操作:
「スタート」→「プログラム」→「アクセサリ」→「ディスクデフラグ」
 - 現在の**Mac**などは、ファイルの保存などのたびにディスクデフラグの処理しているので、ファイルの断片化はなし

HDDとSSDの材料

- **HDD:** アルミニウムやガラスなど

- 円盤の形にして、磁気を持った物質を塗って**HDD**のディスクにする

HDD: <http://www.pasonisan.com/pc-storage/hdd.html>

- **SSD:** 半導体メモリ(フラッシュメモリ)

- 半導体: 電気を通す「導体」という物質と、通さない「絶縁体」という物質の中間の性質を持つ物質

- シリコンやカーボンなど

- メモリ: 記憶媒体

- 「メモリ」だけで使うと、主記憶装置(メインメモリ)のみをさすことが多い
- 他の記憶媒体では、「**USBメモリ**」など、別の言葉とあわせて使うことが多い

SSD: <http://www.pasonisan.com/customnavi/ssd/>

CPUでなぜ命令やデータの保存ができる？

- CPUは、主記憶装置から命令やデータを取り出して処理
- 処理の手順として...

1. 主記憶装置から命令を取り出し
2. 主記憶装置からデータを取り出し
 - 必要に応じて複数回(Ex. 足し算であれば2つのデータの取出しが必要)
3. 処理して結果を算出
4. 結果を主記憶装置に保存

保存ができなければ...

- 2. をしたときに1. の命令の内容が消える
- 2. で2つ目のデータを取り出したときに1つ目のデータが消える
- 3. をするときに1. と2. の内容が消える



CPU内の一時保存場所に保存できる

※詳しくは後期のコンピュータサイエンス2で

1度だけ・何度でも書き込み可能の違いは？

- **CD-R・DVD-RとCD-RW・DVD-RWの違い**
 - データを記録するための物質が違う
 - **CD-R・DVD-R:**「色素」と呼ばれる物質(いくつか種類がある)
 - **CD-RW・DVD-RW:**「アモルファス金属材料」と呼ばれる物質
 - データの記録方法が違う
 - **CD-R・DVD-R:** 色素を焼くことで記録
 - **CD-RW・DVD-RW:** レーザーで熱して、記録の物質内の構造を変えることで記録

解像度[1]

- 原則: 表示内容を、縦横に点を並べて表現し、それぞれいくつの点を使うかを表すもの
 - ディスプレイ: 画面の表示領域の縦横にいくつ点が並んでいるか **XX型 = XXインチ**
 - ディスプレイのサイズ: **XX型 = 対角線のインチ数**
 - 現在の多くのディスプレイ(テレビを含む):
横と縦の長さの比率: **16対9**



Ex1: 12.1型で1920×1080の解像度の場合

対角線²=横²×縦²(三平方の定理, ピタゴラスの定理)で、横:縦=16:9なので...

- 横の長さ≒10.46インチ(約26.57cm)
- 縦の長さ≒5.88インチ(約14.94cm)

解像度 = 横の点の数/横の長さ(縦の点の数/縦の長さ)≒184dpi

※dpi(dot per inch), 1インチあたりの点の数

解像度[2]

- プリンタの場合: 1インチをいくつの点で表すか

解像度4000dpiのプリンタの場合

- 1ドット(点)の大きさ = $1\text{インチ}/4000 = 0.00025 \div 0.00635\text{mm}$
- 1枚の写真(横127mm, 縦89mmのL版)での解像度(横のドット数 × 縦のドット数)
= (横の長さ/1ドットの大きさ) × (縦の長さ/1ドットの大きさ)
 $\div 20000 \times 14016$ (1枚の写真を約280320000個の点で表現)

※点のことを「ドット」や「画素」などとも呼ぶ

Question!

コンピュータの構成(続き)

通信方法(p. 49)

- コンピュータ同士でネットワークを介して通信を行うための装置
 - アナログ回線: モデム, スプリッタ, **ADSL**モデム, **etc.**
 - デジタル回線: スプリッタ, ケーブルモデム, メディアコンバータ, **etc.**
- コンピュータの部品である**NIC**と接続して利用

通信装置[ADSL](p. 49)

- **Asymmetric Digital Subscriber Line**の略
- 非対称デジタル加入者線で、ネットワークを提供するもの
 - 非対称: 上りと下りとで、利用できる通信速度が違うこと
 - コンピュータへ入ってくるデータ: **下り**(ADSLは下りの方が速い)
 - コンピュータから出て行くデータ: **上り**
- 利用のために、家屋の工事が不要
 - アパートなどの借家でも利用しやすい
- 中継局(電話局など)から距離が遠いと、低速になったり通信が不安定になる可能性
- **ADSL**モデムやスプリッタなどの装置が必要

通信装置[FTTH](p. 50)

- **Fiber To The Home**の略
- 光ファイバを一般家庭に設置し、家庭向けのネットワークサービスを提供するもの
 - テレビ, 電話, インターネットなど
- **ADSL**よりも高速
 - 上りと下りの速度が同じ
- 利用のために、家屋の工事が必要
 - アパートなどの借家では、あらかじめ設置されていなければ、利用しにくい
- 中継局からの距離が遠くても、通信が安定
- メディアコンバータなどの装置が必要

拡張カード

- コンピュータに、様々な追加の機能を付加するための部品
 - ビデオカード
 - サウンドカード
 - ネットワークカード
 - **TV**キャプチャカード
 - **etc.**
- なくてもコンピュータは動作するが、ないと人間がうまく使えない、したいことができない、ということも

拡張カード[ビデオカード]

- 「ビデオアダプタ」, 「ビデオボード」, 「**VGA**カード」とも
- コンピュータの画面をディスプレイに表示する装置
 - ビデオカードにより、カラフルな画面が表示可能
 - ビデオカードがなければ、ほぼ白黒の画面
- 画質の性能を決定付ける部品
 - 特に**3次元グラフィック**の表示性能(**2次元**はほぼ同等)

拡張カード[サウンドカード]

- 「サウンドボード」とも
- コンピュータの音声をスピーカーに出力したり、音声をコンピュータに取り込む装置
 - サウンドカードにより、多彩な音が表現可能
 - サウンドカードがなければ、ブザーのような音(ビープ音)のみ
- 音質の性能を決定付ける部品

ネットワーク接続装置[NIC]

- 「**LAN**カード」,「ネットワークカード」,「ネットワークアダプタ」とも
 - **NIC: Network Interface Card**
- コンピュータをネットワークに接続するための装置
 - **NICとADSLモデムやFTTHの壁のネットワークコンセント**に接続して、ネットワークを利用

インタフェース(p. 50)

- **インタフェース**: 装置と装置を接続するときの接続口
- 様々な規格
 - **IDE**: HDDやCD/DVDとコンピュータを接続(内蔵)
 - **Serial ATA**: HDDやCD/DVDとコンピュータを接続(内蔵)
 - 最近主流になってきたインタフェース
 - **USB**: 様々な装置を接続(装置ごとに異なっていたインタフェースを共通化したもの)
 - HDD, CD/DVD, プリンタ, スキャナ, etc.
 - **IEEE1394**: HDD, デジタルビデオカメラ, etc.

インタフェース[**USB**](p. 50)

- 周辺機器を接続するための接続口(ポート)の1つ
- **USB: Universal Serial Bus**
- 様々な外付けの周辺機器を接続するためのポート
 - 従来は、周辺機器の種類によっていくつかのポートが必要
 - キーボード・マウス: **PS/2**
 - プリンタ: パラレルポート
 - モデム: シリアルポート
 - 様々なポートを1つに統一したことは大きなメリット

インタフェース[**IEEE 1394**](p. 50)

- Apple社は「**FireWire**」、Sony社は「**i.Link**」と呼ぶ
- 周辺機器を接続するための接続口(ポート)の1つ
- 様々な外付けの周辺機器を接続するためのポート
 - デジタルカメラや外付け**HDD**など
- コンピュータと周辺機器の接続だけでなく、デジタル機器同士の接続にも利用
 - デジタルカメラとビデオデッキなど

コンピュータの種類

コンピュータの種類(p. 58)

- スーパーコンピュータ
 - 気象学や天文学、流体力学、金融工学など、特定の分野の科学技術計算によく使われる
- サーバコンピュータ
 - インターネットに接続された様々な利用者のコンピュータから依頼をされて、処理をし、処理結果を返すコンピュータ
- パーソナルコンピュータ
 - 個人用の低価格のコンピュータ
- タブレット型コンピュータ
 - iPadなどの薄型コンピュータ

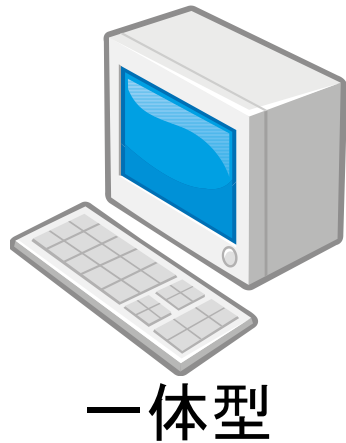
基本的な仕組みは全て同じ!

パーソナルコンピュータ(p. 58)

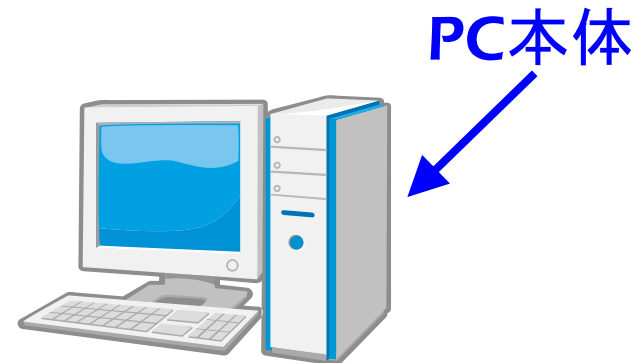
- 個人で使用する小型コンピュータ
 - スタンドアロン(ネットワークに接続しない形)で利用
 - 最近ではネットワークに接続して利用
- 種類
 - デスクトップ型(広い意味)
 - 机の上に備え付けて利用する形態のPC
 - ノート型
 - 持ち運びができて、性能・機能的にデスクトップ型と同等のPC
 - タブレット
 - 携帯型のノートPCよりも小さく、板のような形のコンピュータ

デスクトップ型(p. 58)

- 机に備え付けて利用する形態のPC
 - タワー型
 - PC本体を縦置きにする形態
 - 一体型
 - ディスプレイとPC本体を同じ筐体に収めたもの
 - ※ノート型は、キーボードの下に本体



一体型



タワー型

コンピュータでの情報の扱い方

コンピュータの基本構成

- コンピュータは電気回路で構成
 - 電気回路: 電気が通ることによって動作する様々な部品(電気素子)を電気を通す線で結んだもの
 - **CPU**など、ほとんどの部品は電気回路で構成
- コンピュータは、電気回路に電気が通ることによって様々な命令を処理
 - ある瞬間に、電気回路中のどの線に電気が通ったか・通らなかったかで全ての物事を処理
 - 回路中にたくさんスイッチがあり、ある瞬間でどのスイッチが**ON**でどのスイッチが**OFF**になっていたか、のようなイメージ
 - 人間がコンピュータの動作を考えると、電気が通った線を**1**、通らなかった線を**0**のように数で表現

コンピュータでの情報の扱い方[1](p. 2)

- コンピュータが扱える情報は「0」と「1」のみ
 - ある瞬間で電気が通らなかった線と通った線を0と1として扱って考える
- 大量の「0」と「1」を組み合わせて情報を表現
 - **Ex. 1文字1文字は、0と1の並びで表現**
 - それぞれの物事は、決まった個数の0と1で表現
 - 半角英数文字: 8個
 - 全角文字: 16個
 - etc.
- 様々な情報を「0」と「1」の形(**ビット**)に変換して記録
 - 数, 文字, 画像, 音声, etc.は、全てそれぞれの方法で0と1の並びに変換

コンピュータでの情報の扱い方[2](p. 2)

- 数値は**0と1の並び**で表現
 - 数値を表す**0と1の個数**は、扱い方によっていくつか種類が存在

例えば...

「50」: **110010**

「100」: **1100100**

- **1文字1文字は0と1の並び**で表現

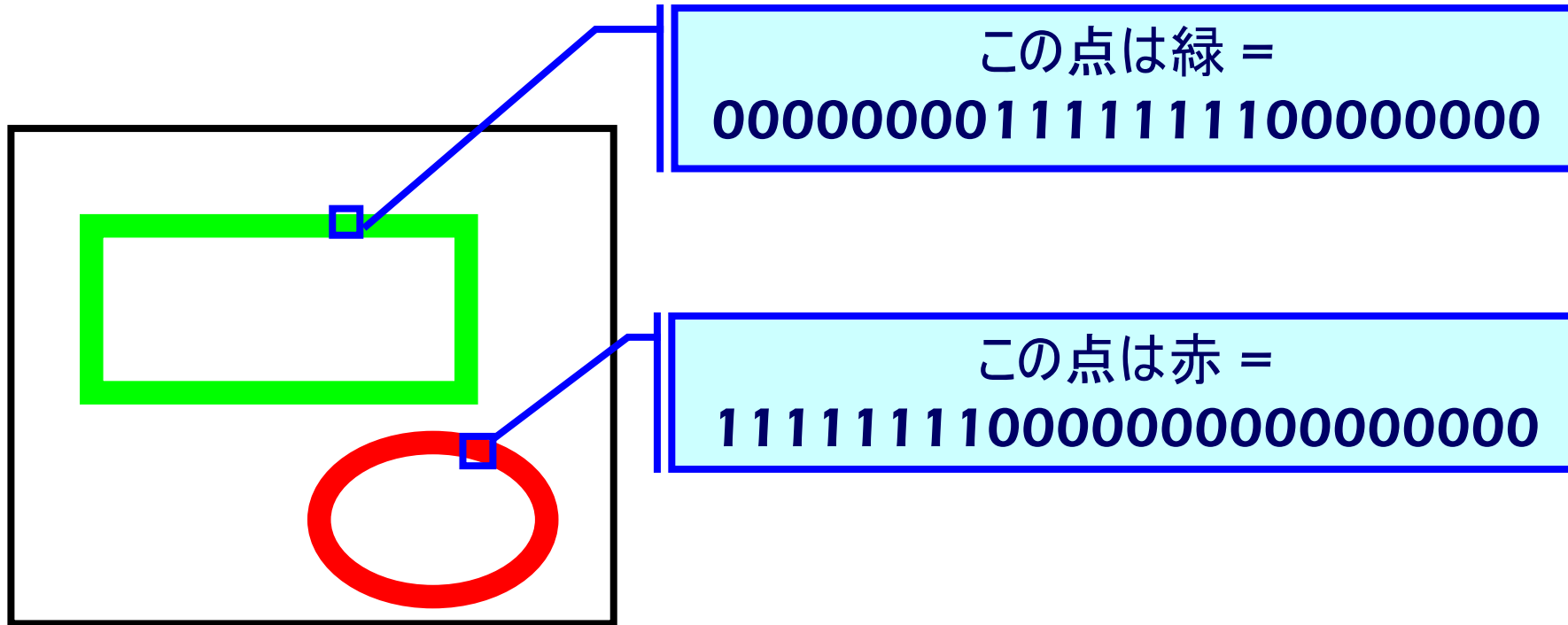
例えば...

アルファベットの「N」: **01001110**
8個の0と1

日本語の「ん」: **1010010011110011**
16個の0と1

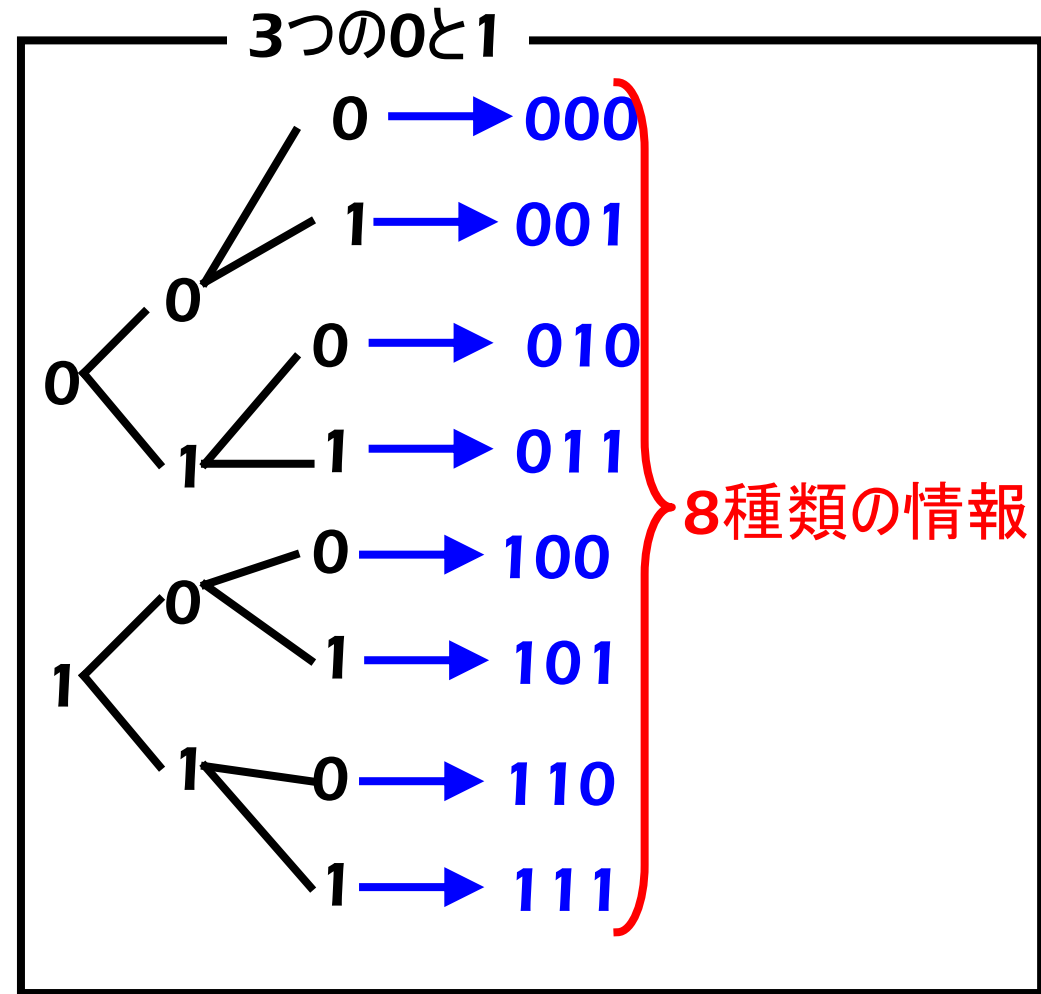
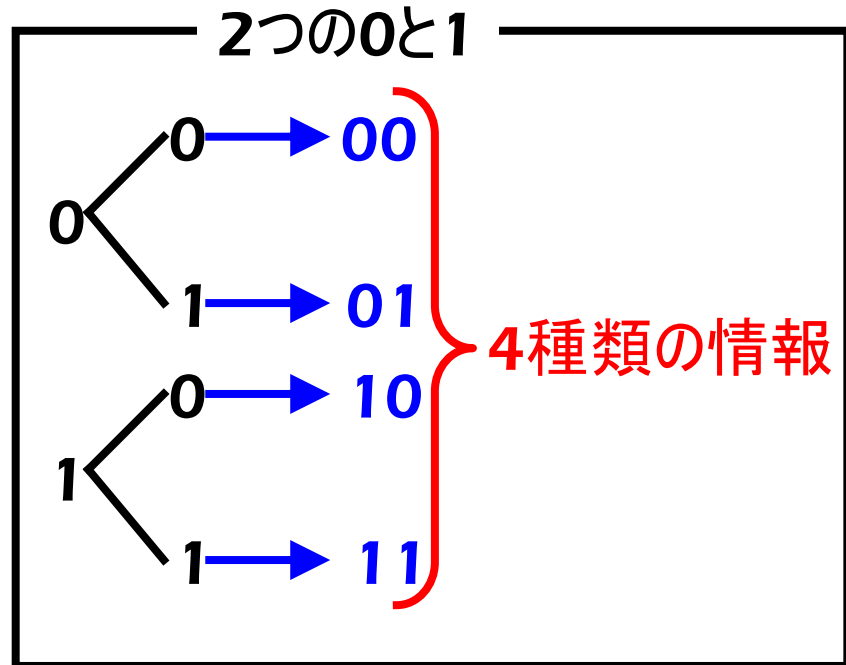
コンピュータでの情報の扱い方[3](p. 2)

- 画像は、コンピュータにとっては点の集まり
 - 1つ1つの点は何色かで絵を表現



ビット[1](p. 4)

- 0と1の組み合わせ



ビット[2](p. 4)

- 2個の「1」と「0」→ 4種類の情報
- 3個の「1」と「0」→ 8種類の情報
- n 個の「1」と「0」→ 2^n 種類の情報
 - 2^n : 2を n 回掛け算する
 - 例: $2^5 = 2 \times 2 \times 2 \times 2 \times 2 = 32$



組み合わせる「0」と「1」の数が多くなれば、
表現できる情報の種類も増える

ビット[3](p. 4)

- **ビット**: 情報を表現する1つ1つの「0」と「1」
 - コンピュータでの情報量の基本単位
 - 情報を表現する「0」と「1」の個数
- **ビット列**: 情報を表現する1つ1つの「0」と「1」の並び

例えば...

「50」: 110010 → 6 ビット

「100」: 1100100 → 7 ビット

アルファベットの「N」: 01001110 → 8 ビット

日本語の「ん」: 1010010011110011 → 16 ビット

情報をどうやって0と1で表す?[1]

- 原則: 個々の情報の内容はすべて同じビット数で表す
- 考え方: 何種類の情報の内容があるか? をもとに、ビット数を決める
 - n ビットとすると、 2^n 個の情報の内容が表せる

Ex1. 1週間の朝食メニュー

洋食・和食・その他・なし、の4種類だった → $2^2=4$ なので、2ビットで表せる

00: 洋食, 01: 和食, 10: その他, 11: なし

※洋食は000、和食は01...のように違うビット数では表さない

そうすると、4/1～4/7の1週間の朝食メニューを表した文書は、コンピュータ的には...

00000100101110



各情報が2ビットなのはわかっているので、各日の朝食が何だったかは、2ビットずつで区切ればわかる



4/1: 洋食, 4/2: 洋食, 4/3: 和食, 4/4: 洋食, 4/5: その他, 4/6: なし, 4/7: その他

情報をどうやって0と1で表す?[2]

Ex1. 天気

晴れ・曇り・小雨・大雨・小雪・大雪、の6種類だった

- $2^2=4$ だと足りない
- $2^3=8$ だと足りる(使っていないビット列の番号があってもOK)

000: 晴れ, 001: 曇り, 010: 小雨, 011: 大雨, 100: 小雪, 101: 大雪

そうすると、ある日の関東(東京・神奈川・埼玉・千葉・群馬・茨城・栃木)の天気を表した文書は、コンピュータ的には... **100011011001001011010**

各情報が3ビットなのはわかっているので、各県のある日の天気が何だったかは、3ビットずつで区切ればわかる

東京: 小雪, 神奈川: 大雨, 埼玉: 大雨, 千葉: 曇り, 群馬: 曇り, 茨城: 大雨, 栃木: 小雨

やってみよう!

- 自分の身近なもの(何でもOK)を0と1で表してみよう!
 - 何ビットで表現できるか?
 - それぞれどういうビット列で表すか?
 - ある一連の情報をビット列で表すとどうなるか?
 - 例の1週間の朝食や関東各県の天気のような感じ

Question!

2進数[1](p. 4)

- **n進数**: 数を**n**個の文字で表す方法
 - **10進数**: 数を**10**個の文字で表す方法(普段使っている数の表現方法)
 - 0, 1, 2, 3, 4, 5, 6, 7, 8, 9の**10**個の文字
 - **2進数**: 数を**2**個の文字で表す方法
 - 0, 1の**2**個の文字

コンピュータ: 「0」と「1」で全ての情報を表現

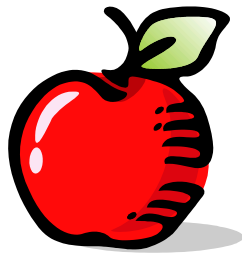
➡ 「**2進数**で情報を表現している」、と言える

2進数[2](p. 4)

- 2進数

- 「0」と「1」だけで全ての数を表現

10進数の「50」 = 2進数で「110010」



を

「りんご」と表現する

「apple」と表現する

表現方法が違うだけ

2進数は、10進数での表現を違う表現にただけ
(数の量などが変わるわけではない)

2進数[3](p. 4)

- 2進数
 - 「0」と「1」だけで全ての数を表現
 - 「2」で繰り上がる、という考え方
 - 10進数: 10で繰り上がる

10進数	2進数
0	00
1	01
2	10
3	11



10進数	2進数
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111



繰り上がり

2進数[4](p. 4)

- n 進数を区別して数を表記する場合: $(\text{数})_n$ と表記
 - 10進数: $(\text{数})_{10}$
 - 2進数: $(\text{数})_2$

$(100)_{10}$: 10進数の百

$(100)_2$: 2進数の100(10進数で4)

10進数を2進数に変換

- 10進数の数は、2進数の表現に直すことができる

2) 10進数の数
2) 商1...余り1
2) 商2...余り2
.
.
.
2) 商n-1...余りn-1
商n...余りn

1. 10進数の数を2で割って商1と余り1を計算する
2. 商1を2で割って商2と余り2を計算する
3. 商2を2で割って商3と余り3を計算する
4.

商が0になるまで繰り返す
※小数の計算はしない

➡ 余りを余りnから余り1の順に左から並べたものが2進数

10進数を2進数に変換(例)

10進数の13を2進数に変換

$$\begin{array}{r} 2 \overline{) 13} \\ 2 \overline{) 6} \cdots \text{余り: } 1 \\ 2 \overline{) 3} \cdots \text{余り: } 0 \\ 2 \overline{) 1} \cdots \text{余り: } 1 \\ 0 \cdots \text{余り: } 1 \end{array}$$



$$(13)_{10} = (1101)_2$$

10進数の50を2進数に変換

$$\begin{array}{r} 2 \overline{) 50} \\ 2 \overline{) 25} \cdots \text{余り: } 0 \\ 2 \overline{) 12} \cdots \text{余り: } 1 \\ 2 \overline{) 6} \cdots \text{余り: } 0 \\ 2 \overline{) 3} \cdots \text{余り: } 0 \\ 2 \overline{) 1} \cdots \text{余り: } 1 \\ 0 \cdots \text{余り: } 1 \end{array}$$



$$(50)_{10} = (110010)_2$$

※矢印の方向に余りを並べる

2進数の桁数

- **10進数**: 普通、「2」や「200」などの数を「02」や「00200」とは表現しない
- **2進数**: 「xx桁の2進数」は、2進数の桁数が「xx」に足りなければ、**2進数の前に「0」をつけて表す**
 - **Ex. 10進数の「2」を6桁の2進数で表せ → 000010**

2進数を10進数に変換

- 単純に...

1. 2進数の各桁の上にそれぞれ「2」を書く
2. 1. で書いた「2」の右肩に、右から0, 1, 2, ...と書いていく
 - $2^0, 2^1, 2^2, \dots$ ができていく

1.

2	2	2	2	2	2
1	1	1	0	1	0



2.

右から左に、0, 1, 2, ...と番号をつける

5	4	3	2	1	0
2	2	2	2	2	2
1	1	1	0	1	0

※ 2^n : 2をn回かけ算する

➤ Ex. 2^3 : $2 \times 2 \times 2 = 8$

2進数を10進数に変換

- 単純に...

3. 各桁の上の「 2^n 」と、それぞれの桁の数をかけあわせる

4. 2. の結果を足し合わせる

2.

2^5	2^4	2^3	2^2	2^1	2^0
1	1	1	0	1	0



3.

2^5	2^4	2^3	2^2	2^1	2^0
×	×	×	×	×	×
1	1	1	0	1	0

||

2^5 2^4 2^3 0 2^1 0



4.

2^5	2^4	2^3	0	2^1	0
-------	-------	-------	---	-------	---

足し合わせる

||

$2^5 + 2^4 + 2^3 + 0 + 2^1 + 0 = 58$

2進数を10進数に変換[4]

- $2^0 \sim 2^{10}$ の数は覚えておくと便利

2のべき乗	10進数	2進数
2^0	1	1
2^1	2	10
2^2	4	100
2^3	8	1000
2^4	16	10000
2^5	32	100000
2^6	64	1000000
2^7	128	10000000
2^8	256	100000000
2^9	512	1000000000
2^{10}	1024	10000000000

やってみよう! [1]

1. 10進数の「25」を2進数に
 2. 10進数の「500」を2進数に
 3. 10進数の「255」を2進数に
 4. 10進数の「135」を10桁の2進数に
 5. 10進数の「200」を12桁の2進数に
 6. 2進数の「001010101010」を10進数に
 7. 2進数の「01111000010」を10進数に
 8. 2進数の「0010000111001」を10進数に
- ビット数も数えてみよう!

※計算方法は、自分でやりやすい方法があれば、それを使って良い

Question!

次回

- **PC組み立て実習**
 - 汚れてもかまわない服装で来ること
 - 遅刻をしないこと