情報・システム工学概論
第1回
導入： 抽象化 と モデル化 そして 仮想化

工学部 電子情報工学科
峯松 信明

講義の目標・概要

- 単純な数値化が難しい対象（人間の言葉・動作・思考・社会的行動 等）に情報技術をどう適用するのか？
- キーになる考え方

  抽象化とモデル化
- 講義スライドは以下のページに置きます

http://www.gavo.t.u-tokyo.ac.jp/~mine/japanese/IT/class.html

minematsu でググって下さい。

抽象 abstraction とは？

事物または表象の或る側面・性質を抽き離して把握する心的作用。（広辞苑第六版）

「金額」を抽き離して考えれば

＝

抽象は捨象を伴う

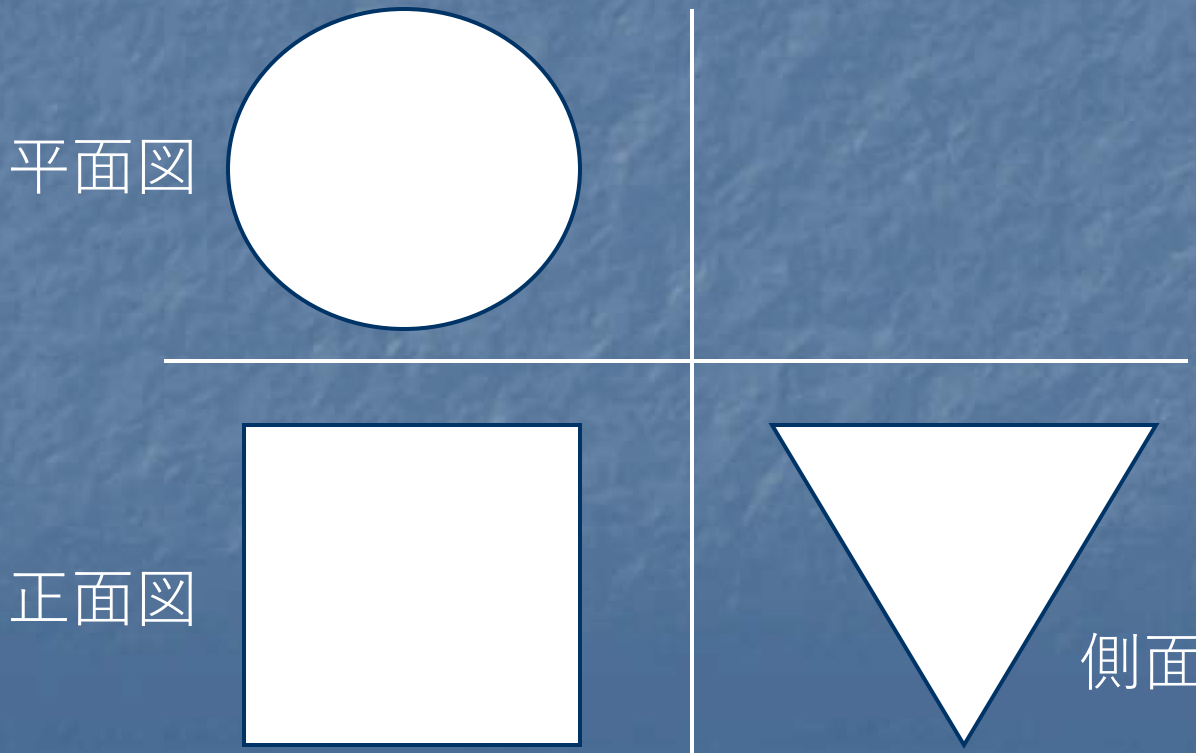
事物または表象の或る側面・性質を抽き離して把握する心的作用。

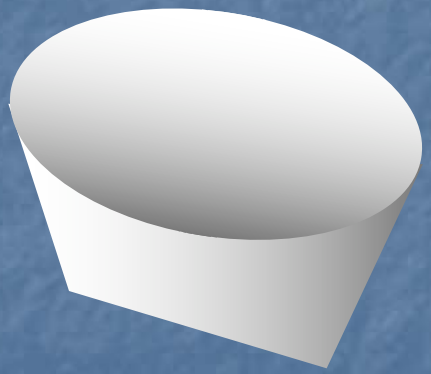
その際自ずから他の側面・性質を排除する作用が伴うが、これを捨象という(広辞苑第六版)。

≠

抽象は射影である

- ものごとの特定の面だけに注目するのが抽象
- 複数の座標軸の一部を捨てて考える ⇒ 捨象





抽象化した結果についての理論

- ３ ｃｍ ＋ ５ ｃｍ ＝ ８ ｃｍ
- ３ｇ ＋ ５ｇ ＝ ８ｇ
- ３円 ＋ ５円 ＝ ８円

} どれも 3+5=8

理論とは？

- 個々の現象や事実を統一的に説明し，予測する力をもつ体系的な知識
  - 三省堂大辞林より
- たとえば
  - If x > y, then, A takes y.
  - If x + y < 10, then, A has to be done again.

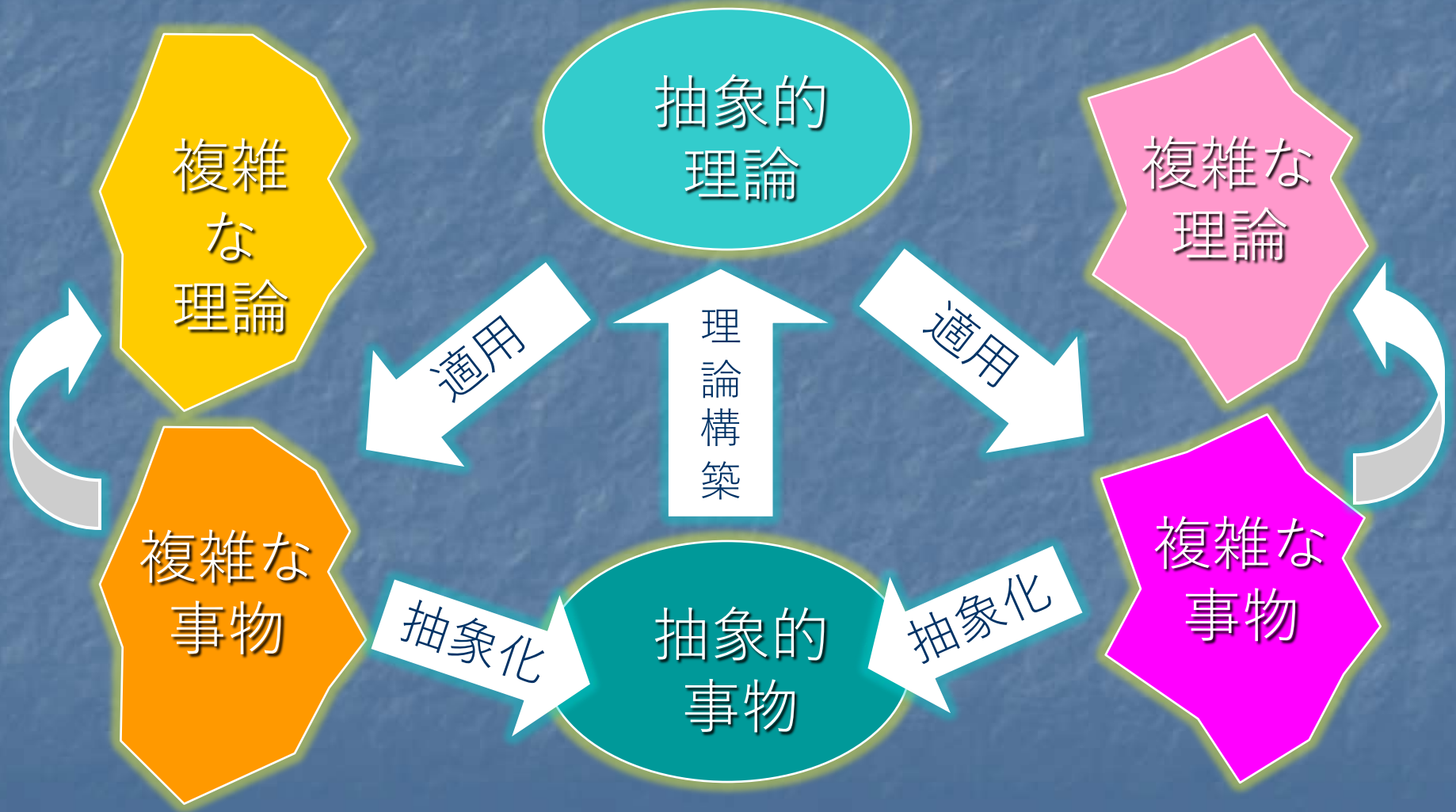
抽象化した理論

適用範囲が広い

∵同じ抽象化ができれば同じ理論を適用可

- 「同じ抽象化」とは？
  - 抽象化した結果について、同じ性質を持つこと
- その性質だけに着目した理論・方法は、同じ抽象化ができるものごとすべてに適用可
- 抽象化で着目した以外の性質には無力

抽象化を用いた理論構築



モデル

- 複雑な事物を簡潔な理論で説明するために、細部を抽象し、簡潔に表したもの
- 必ず捨象されている側面がある
  - 人間 → 三つの数値で表現（B + W + H）
- 逆にいうと、理論が成り立つ世界が、その理論のモデル

モデルと解釈

- モデルは抽象化の結果
- 捨象した部分を適宜補えば現実の事物に
  - モデル： 5 ＋ 3 ＝8
  - 解釈1「トロ五貫とヒラメ三貫の計八貫食べた」
  - 解釈2「5人いたところに3人来て8人になった」
  - 解釈3「5ドルと3ドルで、あわせて8ドルある」

抽象化・モデル化の要点

- 必要な観点を考慮した抽象化が必要
  - 必要な観点を捨象してはならない
    - 人数だけに注目した「員数合わせ」
    - 単位と成績だけに注目した「勉強」
  - 「理論と実際は違う」
    - 当然違う。どう違うか意識して理論を使うこと。
- 不要な観点は捨象した方が理論が簡潔に
  - 「あらゆる点を考慮し…」
    ⇒ 実際は「考えるべき点をすべて考慮し…」

「理論と実際は違う」？

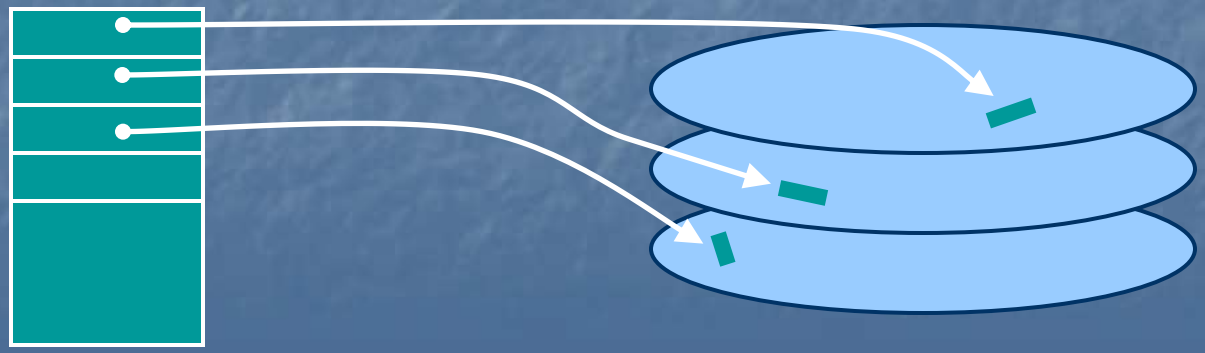
- 理論は細部を抽象したモデル上のもの
  → どこか違っていて当然
- 違うこと自体が問題ではない
- 問題が生じるなら、モデル化が不適切
  - 扱いたい性質を表現できていない
    i.e., 重要な側面を捨象してしまったモデル

「ファイル」という抽象化

- 磁気ディスクは記憶ブロックの並び
  - 記憶ブロックごとに番地（位置を表現する番号）
  - 当初は番地を意識してプログラミング
    × 同じコンピュータで複数の仕事をするとき混乱
- 「ファイル」概念の登場
  - 要は必要なだけのデータが記憶できればよい
  - ファイルごとに適切な数のブロックを割り当て
  - ファイル内の論理ブロック番号と対応付け

ファイル抽象の利点

- 複数タスク間での干渉の回避
- プログラムの可搬性の増大
  - 異なる構成のディスクを持つシステムで再利用可
  - ディスク以外（USBメモリなど）でも同じ記述
- 連続ブロック割り当てが不要に
  - 空いているブロックを適宜使えばよい

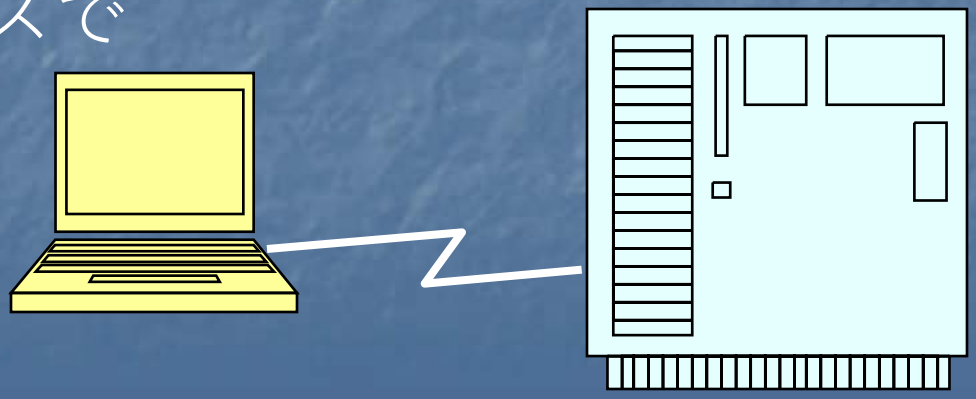


ファイルの論理的構造

ディスクの物理的構造

ファイル抽象の適用範囲拡大

- ネットワークファイルシステム
  - 他コンピュータのディスクも「ファイル」に見せる
  ⇒ 「ファイルサーバ」 も構築可能に
- ディスクではない機器もファイルとして扱う
  - 入出力機器も、ディスクファイルと同じインタフェースで



ストリーム抽象化

- 一般にはファイル読み書きは位置指定可能
  ⇒ ランダムアクセスファイル
- 多くの場合は先頭から順次読み書き
- 順次読み書きしかしないファイルを抽象化
  ⇒ ストリーム： データの入出流口
- 抽象化範囲を絞ってより広い範囲に適用
  - タスク間の通信、ネットワーク通信
  - キーボード、プリンタ、…

メタファー：隠喩

隠喩：修辞法の一つ。あるものを表すのに、
これと属性の類似するもので代置する技法。

[広辞苑第六版]

「〇〇のようだ」などと明示的には言わない。

例：

「人生は旅である」（吉川英治 など）

「人生はキャバレーである」（ライザ・ミネリ）

メタファーの利用

情報インタフェースではメタファーを多用

- File: 切り抜きなどを整理するとじ込み帳
  ∵情報置き場、という観点で
- Folder: 書類挟み
  ∵File をさらにまとめて整理する
- Desktop： 机上 ∵作業場所
- Window: 窓 ∵そこを通して何かが見える

いずれも抽象した機能を共通にもつもの

メタファーの得失

- 「のようなもの」といわないので短く済む
- 既知の事物の類推で機能を理解しやすい
- 喩えられたものに関する概念・用語を借用
  例） Window に対し frame, pane

逆に

- 囚われ過ぎると適切に機能を表現できない

話は変わって…

「バーチャル」と「仮想」

Virtual: 実質上の，事実上の，実際(上)の
（ランダムハウス英語辞典 Windows版 vers. 1.1）

仮想： 仮に考えること。仮に想定すること。
（広辞苑第六版）

まったく意味の異なる言葉で、
本来なら互いの訳語ではありえない

「バーチャル」と「仮想」
たとえば…

Virtual Leader

事実上のリーダ（名目上のリーダは別人）

Virtual Enemy

名目上友好関係にあるが、事実上は敵

仮想敵国

国防計画策定に当たり仮に想定した敵国
実際に敵だといっているわけではない

「仮想」と“Virtual”の結びつきは？

Virtual Memory を仮想記憶と訳してから？

- Virtual Memory：
  実際には存在しない大容量のメモリが、
  存在するのと同様に記述できる仕組
  - 実際はメモリとディスクの内容の交換で実現
- これに「仮想記憶」という訳を与えた
  - Virtual：「存在するのと同様」に力点
  - 仮想：「実は存在しない」に力点

“Virtual Reality”と「仮想現実」

- Virtual Reality：（実は現実ではないが）現実と同様の効果を持つもの
- 「仮想現実」というと「現実ではない」を強調した語感だが、本来その意味ではない
- 現実と間違えようもないちゃちな装置は、本来はVirtual Reality ではない

Virtual の前提

「事実上○○である」ためには

- ○○の重要な性質を備えている

  例) リーダはグループの行動方針を決める

  ⇒ 方針を決めるのが Virtual Leader

- その重要な性質が何なのかが問題

  ⇒ 目的によって異なる

  ⇒ 目的に応じた抽象化

仮想 と Virtual

コンピュータ屋が使う意味では

- 本来の事物そのままではない、が
- 本来の事物の ある重要な性質を持つ

例)

- 仮想商店街： 物品の売買ができる
- 仮想記憶： 大容量メモリを使える
- …

ソフトウェア構築における抽象化と仮想化

抽象化による汎用ソフトウェア部品の構築

- 望ましい特定の性質を抽象
- その性質を持つようにソフトウェアを構築

仮想化による汎用化

- 機能の一部を抽象概念として再構成
- その機能を持つ仮想存在として扱う

⇒ 適用範囲の広い概念、汎用性の高いソフト

インターネットの仕組

- 情報を複数のパケットに分けて少しずつ送る
- 行き先だけ指定、経路はまちまち
- 途中で失敗も
- うまく届かなければやり直し

インターネットは基本的に低品質

- 通信経路上の故障・混雑によるパケット損失
- 複数経路が原因の到着順の入れ替わり

こうした問題のないネットワークは高コスト

- 求められる機能を抽象化すると…

「送ったデータは送った順にすべて届く」

⇒ これが実現されれば事実上高信頼NW

低信頼ネットワークを高信頼に見せたのが
インターネットの成功の秘訣

伝送制御プロトコル TCP

必ず送った順に届く仮想ネットワークを実現

- 実際にはパケット損失や順序の前後も
- パケットに番号をつけ、受信してもひとつ前のパケットを受信するまでは知らせない
- 来るはずのパケットが長時間来なければ、送信元に再送信を要求

⇒ 高信頼ネットワークのように見える

ネットワークの機密保持

- 機密保持が重要なデータ通信用ネットワーク
  - 傍受されない専用のネットワーク ⇒ 高コスト
- 求められる機能を抽象化すると…
  - 通常のネットワーク通信ができること
  - 通信内容を秘匿できること

⇒ これらの機能を実現できれば、
物理的に専用のネットワークでなくてもよい

仮想プライベートNW　VPN

- パケットを暗号化してから通常の通信
- 傍受されても解読できないので、
  通信内容は秘匿される
- 暗号化・復号化はネットワークシステムが行う
  ∴利用する側から見ると通常のNW通信

⇒ 専用ネットワークを構築したのと事実上同様

仮想記憶

- 小容量のメインメモリがあたかもたっぷりあるかのうように見せる技術
- 実際にはメモリと磁気ディスクなどの間で内容を交換（スワップ）することで実現

メモリの特性

コンピュータのメモリは

○ 番地を指定してすぐ読み書きできる
　∴ 高速な動作が可能
　∴ プログラムの記述が簡単

× 記憶容量あたりでは比較的高価
　∴ あまり大容量のメモリはコスト高

磁気ディスク等の特性

二次記憶装置（磁気ディスク等）は

○ 記憶容量あたりでは低価格
　∴ 大容量を低コストで実現できる

× 特定の内容を読み出すのは遅い
　∴ 直接記憶装置として使うには動作が遅い

× 読み書きには入出力操作が必要
　∴ プログラミングが面倒

△ 連続したデータの読み書きはかなり速い

実現したいのは

- 大容量の記憶を低コストで
- 記憶の読み書きを高速に
- メモリを使うのと同様のプログラミング

でもそんなに都合のよい記憶装置は
実際には存在しない

メモリアクセスの特性

多くのプログラムの持つ性質

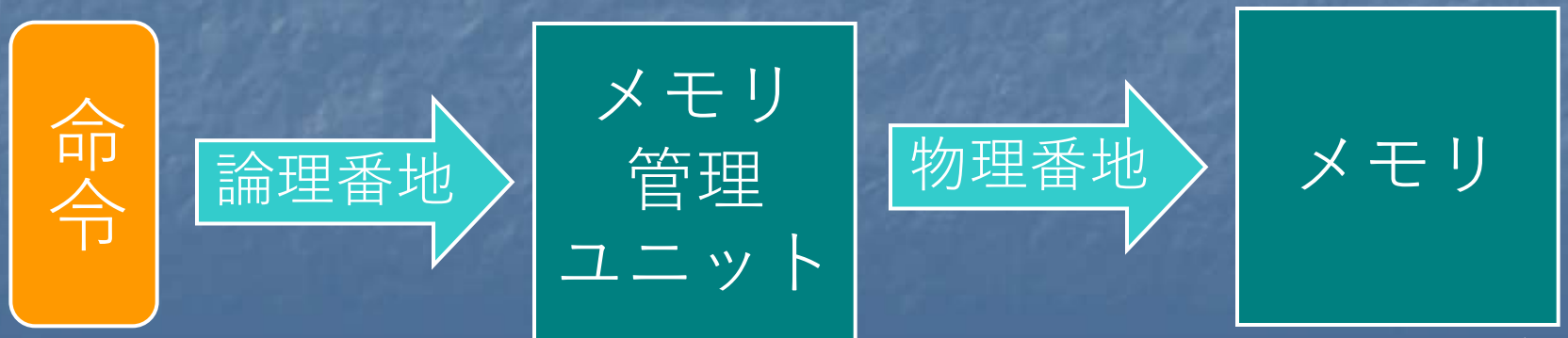
- 最近使った番地を、またすぐ使うことが多い
  時間的局所性
- 最近使った番地の近くを使うことが多い
  空間的局所性

これらの特性を持ったプログラムに対して、
「指定番地をすぐ読み書き」できればよい

⇒ 最近使った番地の付近はメモリ上に
他は磁気ディスクに置いておけば良い

仮想記憶の仕組： アドレス変換

- 命令で指定するメモリ番地は論理番地
- 論理番地と実際の番地の対応付け装置（メモリ管理ユニット）
- 命令指定の論理番地を物理番地に変換



仮想記憶の仕組： ページング

- 論理番地の範囲は物理メモリ量を超えてよい
  ⇒ 対応する物理番地のない論理番地もある
- アドレス変換例外とその処理
  - 指定論理番地に対応する物理番地がなければ
    オペレーティングシステム (OS) に通知
  - OS がディスクとの間でページを適宜交換
  - 交換後に元のプログラムを再開
  - プログラムには何もなかったかのように見える

Virtual！

仮想記憶の仕組： なぜうまくいく？

- メモリアクセスの空間・時間局所性
  ⇒ 多くの場合、アクセスする論理番地に
  対応する物理番地が既にある
- ディスクの特性
  ⇒ 連続するデータの読み書きは比較的速い
- マルチタスク
  ⇒ ページ入替え中にも別の仕事があるので、
  ディスクの動作間も CPU は有効利用

記憶仮想化の拡張

- ページ入れ替えはネットワーク越しでもよい
  ⇒ ディスクのないコンピュータ
- 複数のコンピュータでページを共有
  - Virtual Shared Memory
    複数のコンピュータで事実上メモリを共用
  - 自分だけが持つページは黙って書き込んでよい
  - 他のコンピュータにもあれば書き込みを通知

コンピュータの仮想化

- コンピュータは OS の管理下で動作
  - さまざまなサービスの提供(仮想化した機器等)
  - 複数のタスク間の競合の解決
  - ひとつのコンピュータに OS はひとつ
- 特定 OS でしか動かないソフトウェアも多い
  - 特定 OS の機能を利用
  - 特定 OS のインタフェース （API） を使用

⇒ ひとつのシステムで複数の OS を使いたい

エミュレーションによる実現

- コンピュータの動作をシミュレートすればよい
  - OS も所詮は命令列を実行しているだけ
  - 個々の命令を解釈して実行をシミュレート
  - 入出力機器は仮想化してシミュレート
    - ディスクはファイルで
    - 端末画面はウィンドウで

⇒ これでは非効率すぎる

OS 機能以外は普通に実行すればよいのに

OS による保護機能の実現

OS がコンピュータを管理できるのは...

- 入出力などの命令は OS だけが実行できる「特権命令」
  ⇒ 利用者のできることを制限
- 普通のプログラムが特権命令を実行すると
  ⇒ OS が呼ばれてエラーを報告

この機構をうまく使えないか

仮想マシン

OS も含めて通常のプログラムとして実行

- 普通の命令は普通に実行できる
- 特権命令を実行しようとするとエラー

⇒ このエラーを捕らえて、
　入出力などをシミュレートすればよい

実行速度は直接 OS を実行するのに近い

あたかも別の計算機があるも同様

今日の講義のまとめ

- 効率的な理論・方法の構築に抽象化が重要
- 抽象化はある側面に注目、他は捨象する
- Virtual とは「実際上そうである」こと
- なにが「実際上」なのかは着目点次第
- ソフトウェアは抽象化と仮想化が鍵

スライドは以下の Web ページ中に置きます

http://www.gavo.t.u-tokyo.ac.jp/~mine/japanese/IT/class.html

minematsu でググって下さい。

成績評価の方法

- 出席回数
  - 出席点は結構大きい・・・・
- 学期末レポートの提出（原則電子メールで）
  - 全6教員のうち3教員以上の分について提出
  - 課題は担当教員ごとに出題（今回については なし）
  - どの回を選んで提出するかは自由
  - 調査は大事だが、引用は引用と明示すること
  - コピペをオリジナルかのように偽るのは絶対ダメ

http://www.gavo.t.u-tokyo.ac.jp/~mine/japanese/IT/class.html

minematsu でググって下さい。