

# システム管理論

-- OSの理解 --

栗野 俊一

講義内容の静止画・動画での撮影、及び SNS 等への転載を固く  
禁じます

2024/04/16 システム管理

# 伝言

---

## 私語は慎むように !!

### □ 席は自由です

- できるだけ前に詰めよう
- コロナ対策のために、ソーシャルディスタンスをたもとう

### □ 色々なお知らせについて

- 栗野の Web Page に注意する事

<http://edu-gw2.math.cst.nihon-u.ac.jp/~kurino>

- google で「kurino」で検索

# お詫び: 第01回の動画について

---

## □ 第01回の動画について

- 講義動画は作成した ( 月曜日の方は、目にしているはず.. )

  - ▶ 削除されてしまった(とりあえず、栗野の責任ではないが..)

- 現在は、昨年度(2023年度)版を公開している

  - ▶ 今年度はこれで行く

## □ 教訓(色々..)

- 人間は失敗する生物/天災は忘れた時にやってくる

  - ▶ 保険の考え方は重要

  - ▶ バックアップは ? : ユーザはサービスを過信してはいけない

- 「情報システム」の怖さ: 簡単に色々な事ができてしまう(拡大構造)

  - ▶ 便利 ? : 情報システムの最も良い性質

  - ▶ 危険 ? : 情報システムを扱う上で、念じて置く必要がある性質

- End User の資産の損失

  - ▶ (自分ではなく..)他人の資産を与るという認識

# 前回の復習

---

システム管理論

前回の復習

講義内容の静止画・動画での撮影、及び SNS 等への転載を固く禁じます

# 前回の復習

---

## □ 前回の復習

### ○ 初回だったのでガイダンス：講義を受講するかどうかを判断する

- ▶ 話者の紹介 (「kurino」で検索 / 一昨年、理工学部から経済学部に移った)
- ▶ 講義の方針 (資料の参照方法 / 講義の方針 / 評価方法)
- ▶ 講義の目的：サーバやネットワークシステムを管理する能力を身に付ける

### ○ 講義：システム管理とは

- ▶ システムとは？：機能を持つモジュール(部品)を複数組み合わせサービスを提供する物
- ▶ システム管理：サービスの構築や維持を行うためにシステムの面倒をみる
- ▶ システム管理の例：システム構築 / システムの保守

# 今回の概要

---

システム管理論

今回の概要

講義内容の静止画・動画での撮影、及び SNS 等への転載を固く禁じます

# 本日(2024/04/16)の予定

---

## □ 本日(2024/04/16)の予定

- 講義

- ▶ OS とは

- 実習

- ▶ PowerShell の利用

- ▶ WebClass の利用

# 今日(2024/04/16)の目標

---

## □ 今日(2024/04/16)の目標

### ○ 講義

▶ 「OS」の一次理解 ( OS の役割/必要性, OS と他のモジュールの関係 )



# 本日の課題 (2024/04/16)

---

## □ 前回の課題

- 初日なのでなし

## □ 今週 (2024/04/16) の課題

- 課題-02: 画面にメッセージを出力するバッチファイルの作成

# コンピュータの構造(ハードウェア)

---

システム管理論

## コンピュータの構造(ハードウェア)

講義内容の静止画・動画での撮影、及び SNS 等への転載を固く禁じます

# コンピュータの構造

---

## □ コンピュータの構造

### ○ CPU(MPU) : 計算や制御等、プログラムを実行する主体

▶ CPU がメモリから命令を読み込み、実行する (ボトルネック)

### ○ RAM : CPU が参照操作するデータを保存する

▶ CPU が実行するプログラムも保存されている (ノイマン型)

▶ 電源を消すと情報は消えてしまう ( cf. ROM )

### ○ 外部記憶装置 : 電源を消しても消えない記憶領域

▶ RAM (Main Memory) に対して、大容量だが、遅い

### ○ I/O : Input/Output

▶ コンピュータの外との情報のやり取りを行う (周辺器機)

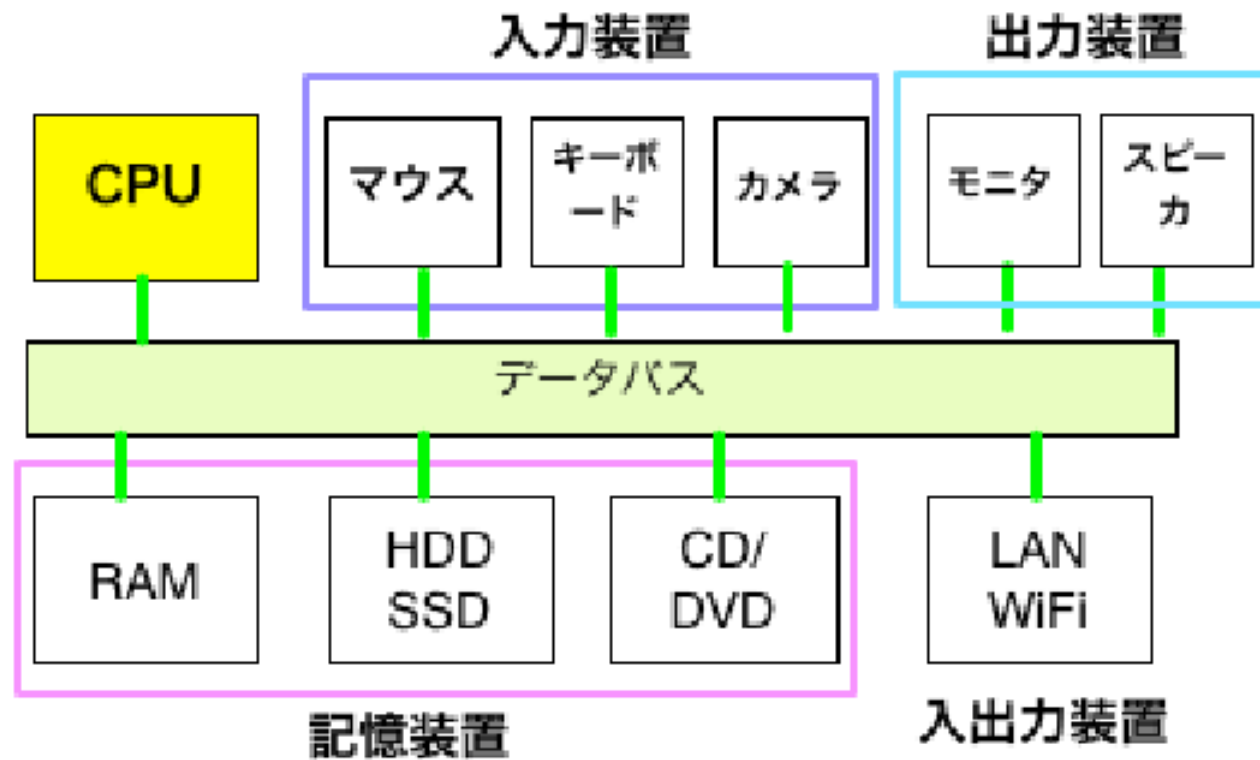
### ○ バス

▶ CPU と他の装置とのデータのやり取りをする場所

# コンピュータアーキテクチャ

---

## コンピュータアーキテクチャ



## コンピュータアーキテクチャ

# OS とは

---

## システム管理論

# OS とは

講義内容の静止画・動画での撮影、及び SNS 等への転載を固く禁じます

# OS とは

---

## □ OS (Operating System) とは ?

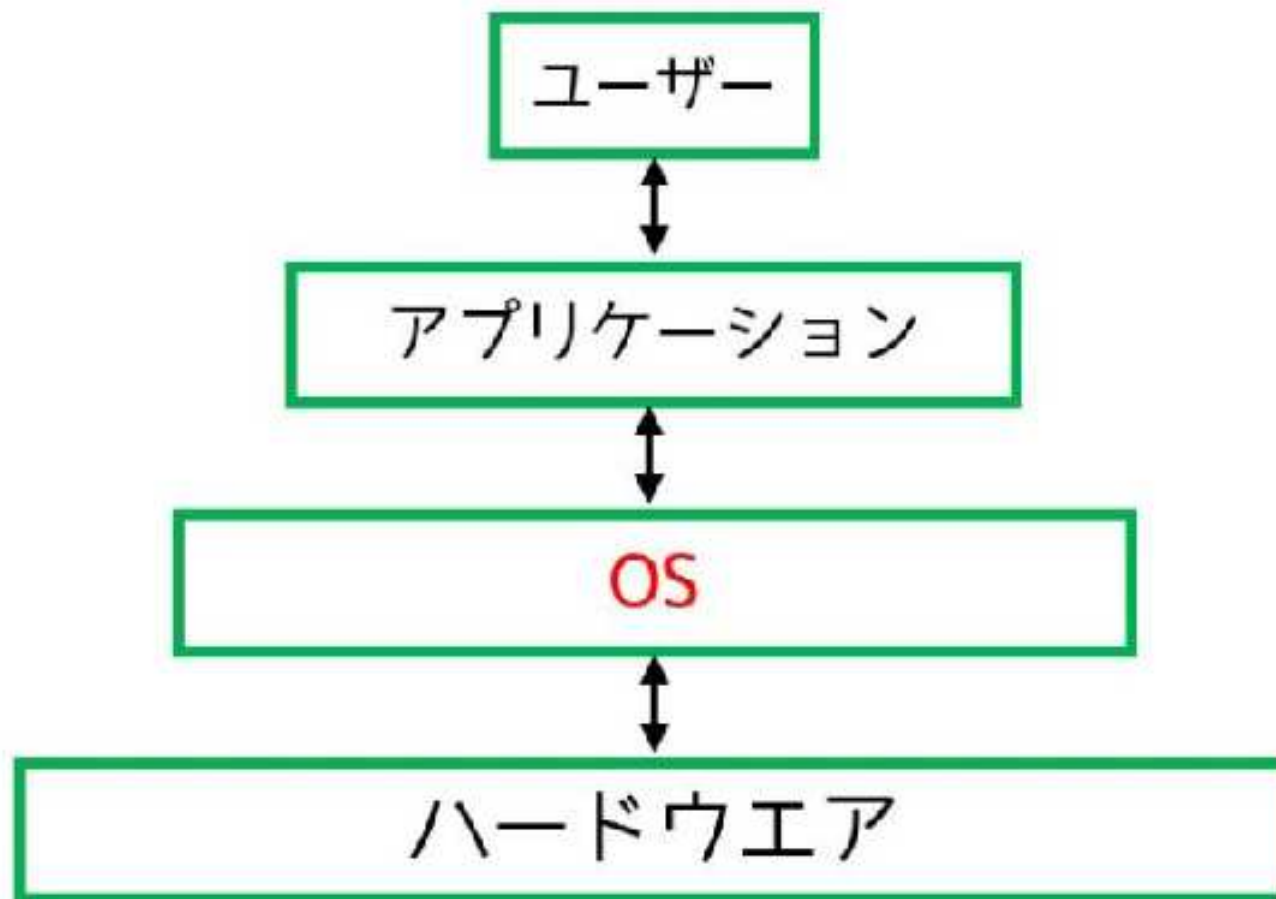
- ハードウェアの資源を管理し、アプリケーションに対するサービスを提供する
- OS の例
  - ▶ PC 用 : MS-Windows, unix (linux, MacOS)
  - ▶ 携帯電話用 : iOS, android, MS-Windows
  - ▶ ゲーム機 : 個々のゲーム毎に OS が提供されている

## □ OS の役割

- 計算機資源の管理 ( CPU, Memory, 外部記憶装置, 周辺器機, etc.. )
  - ▶ 資源を管理し、色々な機能を実現 ( マルチタス, 仮想記憶, ファイルシステム, Window System )
- アプリケーションプラットフォーム ( System Call )
  - ▶ 資源の抽象化/仮想化し、利用方法(API)の共通化を提供
- 利用者との仲立 ( shell )
  - ▶ 利用者に、操作手段を提供する

# システムの階層図

---



システムの階層図

# Unix OS のアーキテクチャ

---

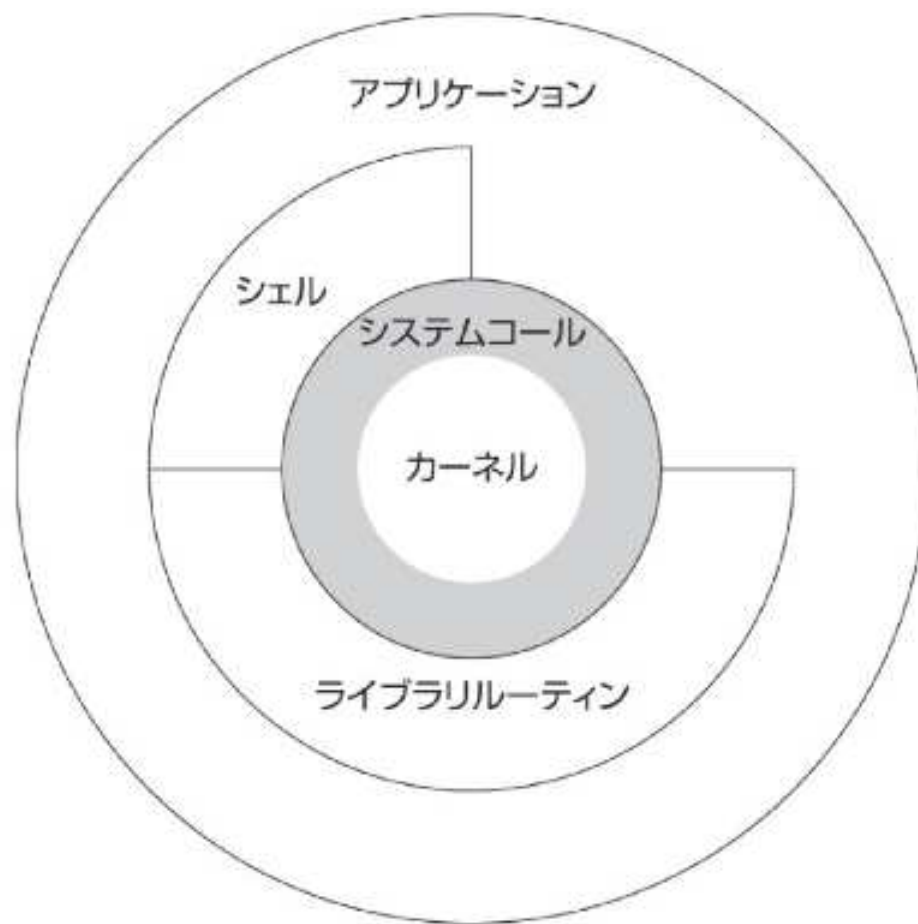


図1.1 : Unix オペレーティングシステムのアーキテクチャ

Unix OS のアーキテクチャ



# プラットフォーム(共通化)

---

## □ I/F (Interface : 境界) の役割

### ○ I/F とは ?

▶ 異なる物の間にある境界の事

### ○ I/F の共通化

▶ I/F が共通なら、組み合わせる事ができる

▶ 例 : プラグとコンセント, ネジとナット, 机と答案用紙

▶ I/F が共通じゃないと、組み合わせができない

▶ 例 : 日本と中国の電源, ミリ螺子とISO螺子, 傷があつて凸凹

## □ $M \times N \rightarrow M+N$ 法則

### ○ 二種類 ( 一方が $M$ 通りで、他方が $N$ 通り ) の組み合わせ

▶ I/F が共通でない場合 :  $M \times N$  の組み合わせを作る必要がある

▶ I/F が共通の場合 :  $M + N$  通りで済む

## □ プラットフォーム

### ○ 共通な I/F を提供する物

## □ OS : ハードとアプリの I/F になっている

### ○ アプリのプラットフォームになっている

# unix OS

---

## □ unix OS (ユニックス) とは

- 1969年、ケン・トンプソン、デニス・リッチーが開発 (Multics に対する反省)

- ▶ Open な OS の一つ ( vs MS-Windows )

- ▶ 色々な派生 unix link OS を生んだ

## □ Linux

- Linus の作った PC 上の unix os の kernel の実装

- ▶ サーバ OS としてよく利用される

## □ デイトリビューション

- Linux Kernel に色々なソフトを追加して、OS としての体裁を整えたもの

- ▶ Ubuntu, Android

# ファイルシステムとパス名

---

## システム管理論

# ファイルシステムとパス名

講義内容の静止画・動画での撮影、及び SNS 等への転載を固く禁じます

# ファイルシステム

---

## □ ファイルシステムとは

- 記憶装置上の情報の塊を「ファイル」という論理単位で扱えるようにする仕組み

- ▶ cf. 外部記憶装置：大量のデータが連続して記録されている

## □ ディレクトリ(階層)構造

- ディレクトリ：複数のファイルをまとめたもの (フォルダ)

- ▶ フォルダの中にフォルダを入れる事ができる

- ▶ 全体として大きな木構造になる

- ルートディレクトリ

- ▶ 木の根っこの部分

## □ カレントディレクトリ

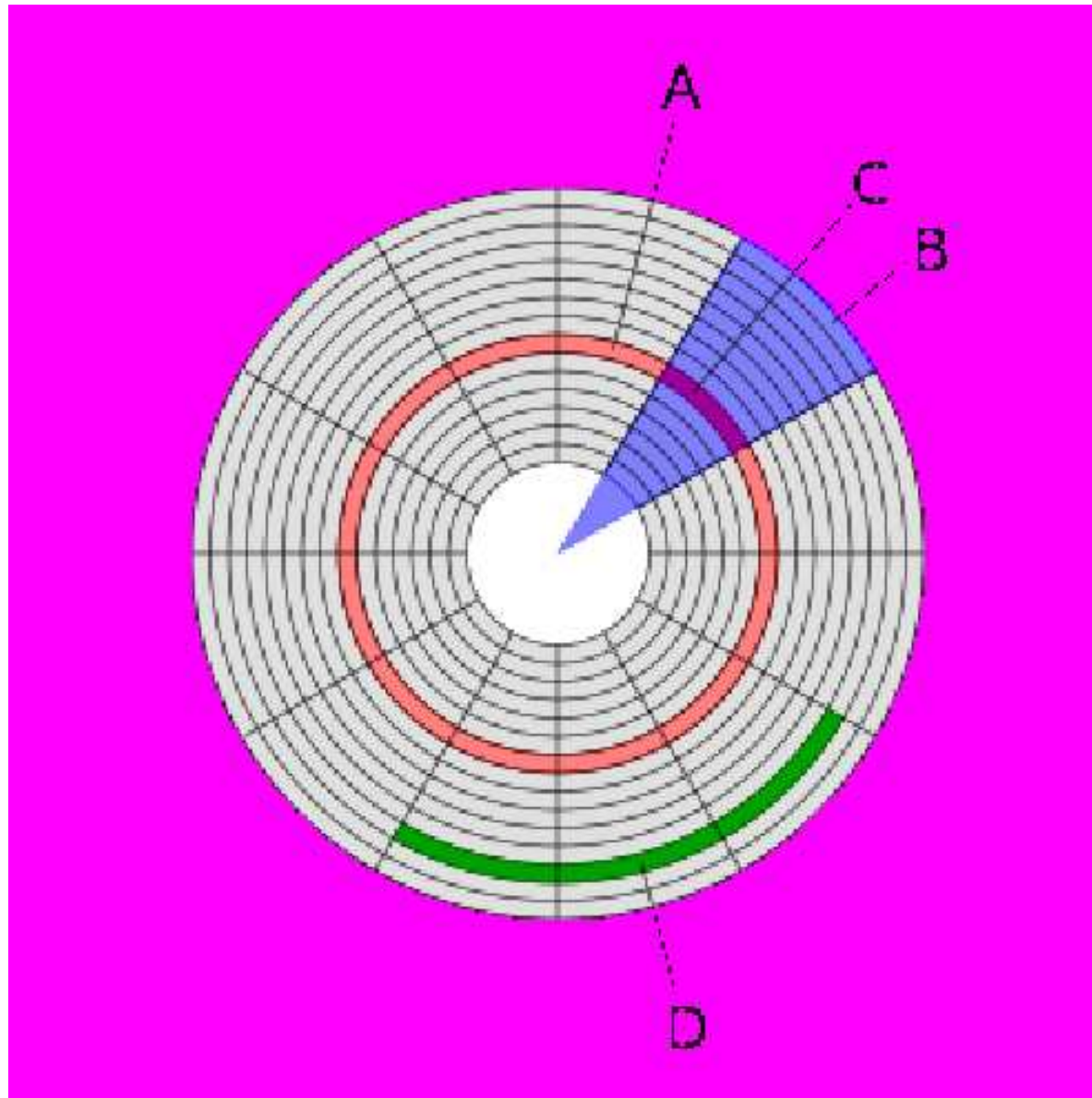
- 現在着目しているディレクトリ

## □ パス名：ディレクトリ名とファイル名を並べたもの

- 相対パス名：カレントディレクトリから対象のファイルを指定した名前

- 絶対パス名：ルートから対象のファイルを指定した名前 (/で始まる)

# ディスクの構造



ディスクの構造

A -- トラック / B -- セクタ / C -- トラックセクタ / D -- クラスタ

# 実習: PowerShell とバッチファイル

---

システム管理論

## 実習: PowerShell とバッチファイル

講義内容の静止画・動画での撮影、及び SNS 等への転載を固く禁じます

# 実習: PowerShell とバッチファイル

---

## □ CUI の体験

### ○ PowerShell : Windows に用意されている CUI I/F

▶ GUI とは別の形で、ファイル操作が可能

### ○ PowerShell で利用できるコマンド

▶ echo メッセージ : メッセージを出力

▶ copy ファイル1 ファイル2 : ファイルのコピー

▶ ren ファイル1 ファイル2 : ファイルの名前の変更

▶ del ファイル : ファイルの削除

### ○ バッチ

▶ 命令を並べたもの : 実行すると並べた順に一度に実行可能

# 実習: 課題-1

---

## □ 課題-1

- 内容: 画面にメッセージを出力するバッチファイルの作成

- 手段:

  - ▶ メモ帳で、sysad-02-NNAANNN.bat (NNAANNN は学生番号) を作成

  - ▶ その中に、「echo メッセージ」の形のテキストを書き込む

- 確認:

  - ▶ PowerShell で、バッチファイルを実行して、おもった通りに動く事を確認

- 提出:

  - ▶ WebClass でファイルをアップロードして提出



おしまい

---

システム管理論

おしまい

講義内容の静止画・動画での撮影、及び SNS 等への転載を固く禁じます