

集中講義令和8年2月24日～26日

コンピュータシステム入門



Day 3: オペレーティングシステム（OS）の役割と仕組み

講師 陳 オリビア（大学院システム情報科学研究院）

TA GPT-5 Thinking (OPEN AI)

Gemini 2.5 pro (Google)

今日のスケジュール

	時間帯	モジュール	タイプ
午前中	10:00 ~ 11:00	オペレーティングシステムの役割と仕組み	講義
	11:00 ~ 11:15	Coffee Break	
	11:30 ~ 12:30	未来のコンピュータ技術と応用の展望	講義
	12:30 ~ 13:30	Lunch Break	
午後	13:30 ~ 16:00	LINUX操作演習	演習

モジュール7：オペレーティングシステム（OS）の役割と仕組み

01

なぜOSが必要か

03

人間との対話方法

02

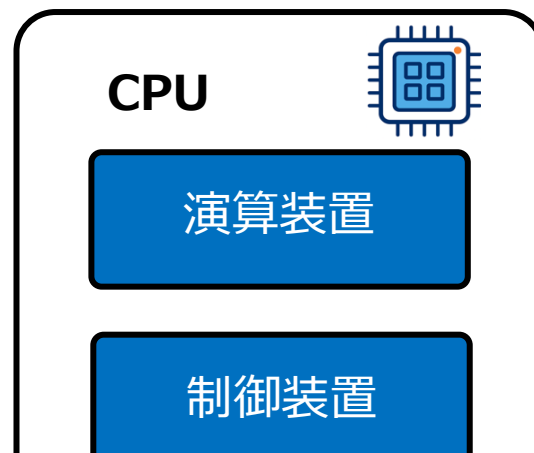
OSの機能

04

OSの種類と歴史

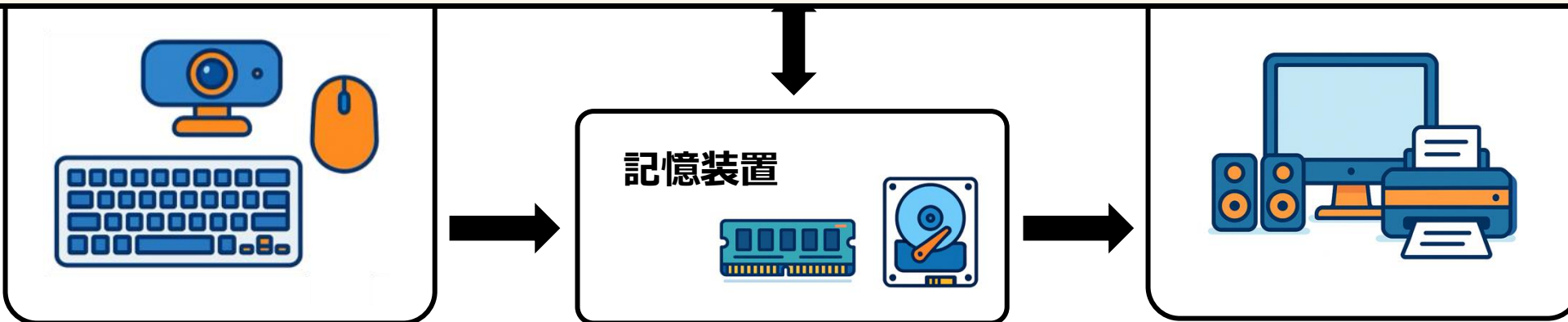
ここまでの復習：コンピュータの5大装置

- 入力：人・環境→データ
- 出力：データ→人・物理世界



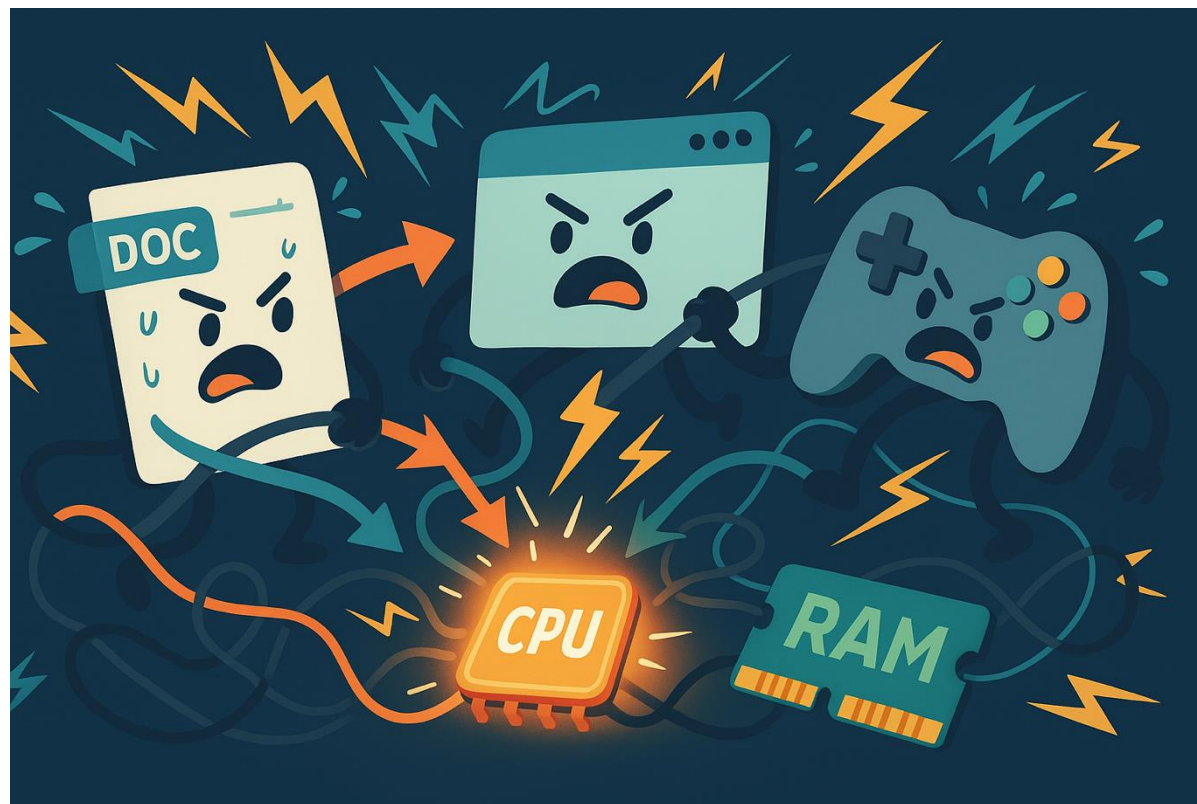
- **CPU（制御+演算）**：命令を解釈し、計算し、全体を制御
- **メモリ（主記憶）**：作業中のデータを一時的に展開
- **ストレージ（SSD/HDD）**：長期保存（非揮発性）

しかし、これらの部品は、ただ存在するだけでは何もしてくれない！

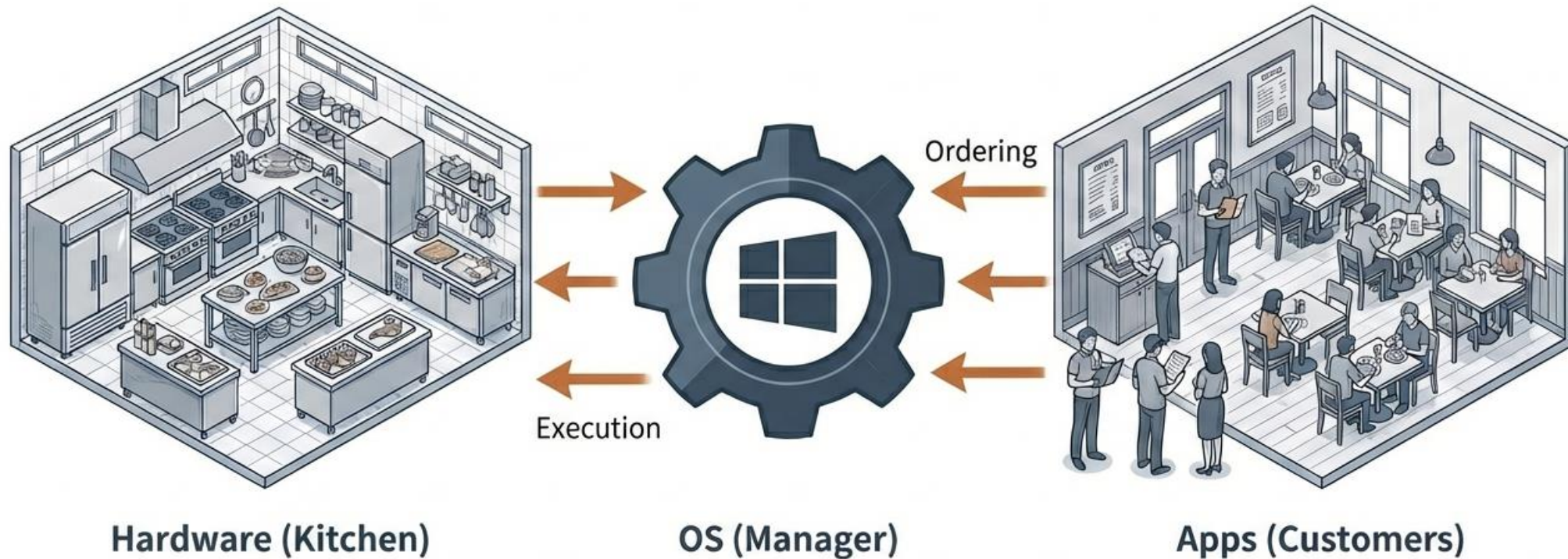


もしOSがなかったら…？

- 誰がCPUを使う順番を決める？
- 誰がメモリの場所を割り当てる？
- 誰がファイルを管理する？
- 誰がプリンターと対話する？



OSとは『偉大なる管理人』である



- **Hardware (Kitchen)**: CPU, Memory, Storage. 能力はあるが、自律的に動けない。
- **OS (Manager)**: 資源を『誰が・いつ・どう使うか』を決定・調整する。
- **Apps (Customers)**: 資源を注文するが、調理（処理）の段取りはOSに任せる。

モジュール7：オペレーティングシステム（OS）の役割と仕組み

01

なぜOSが必要か

03

人間との対話方法

02

OSの機能

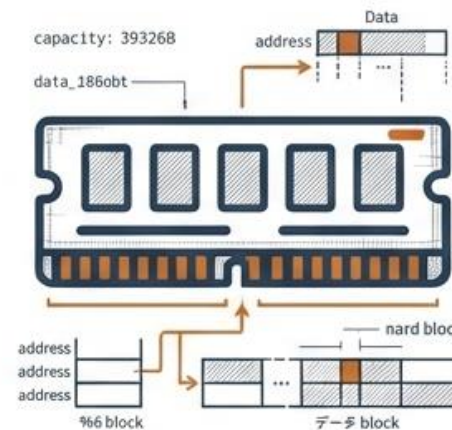
04

OSの種類と歴史



Process Management (プロセス管理)

マルチタスクの指揮官



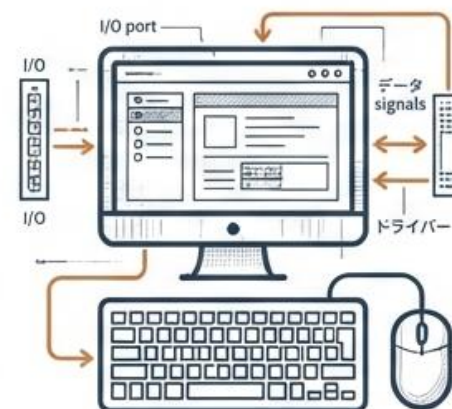
Memory Management (メモリ管理)

限られた空間の公平な分配



File System (ファイルシステム)

データを『本』として整理する



Device Management (デバイス管理)

周辺機器との通訳

プロセス管理：プログラムの生涯



Context Switch: 保存と復元

スケジューリング・アルゴリズム



Round Robin (RR)

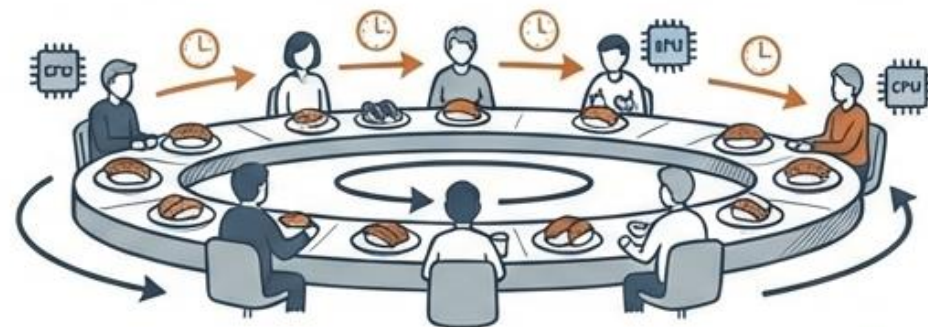
タイムスライスで均等に配分。現代OSの基本。

Priority Scheduling

緊急度（優先度）が高いタスクを最優先する。

FCFS (First Come, First Served)

到着順。前の人が遅いと全員待たされる（非効率）。

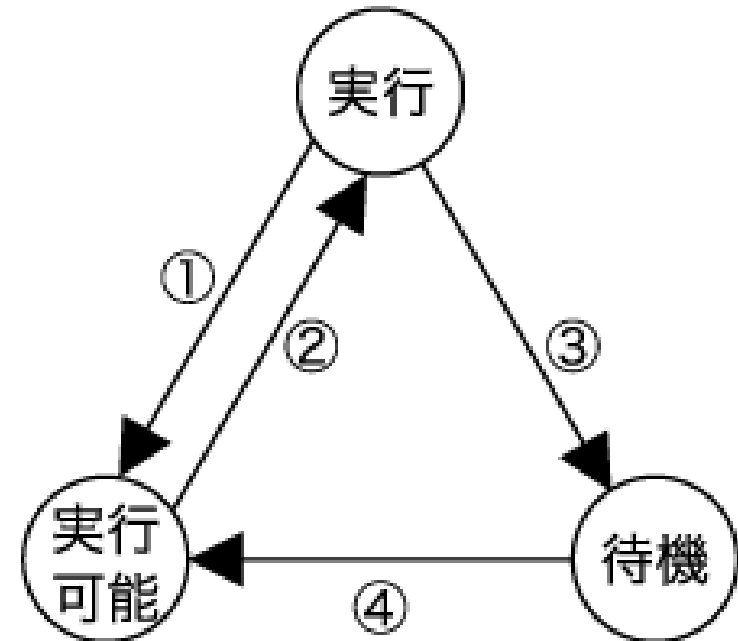


The 'Scheduler' decides the logic of the CPU queue.

機能① プロセス管理

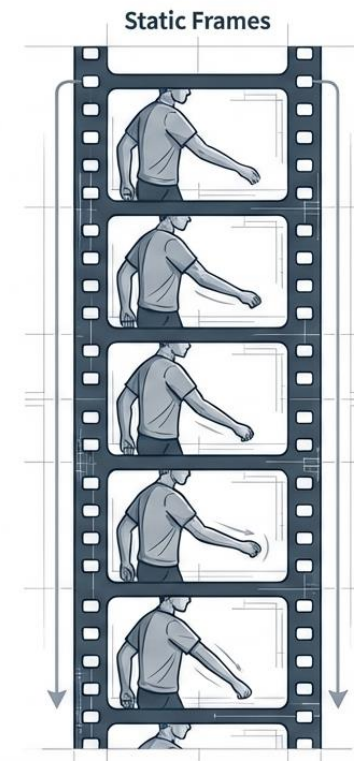
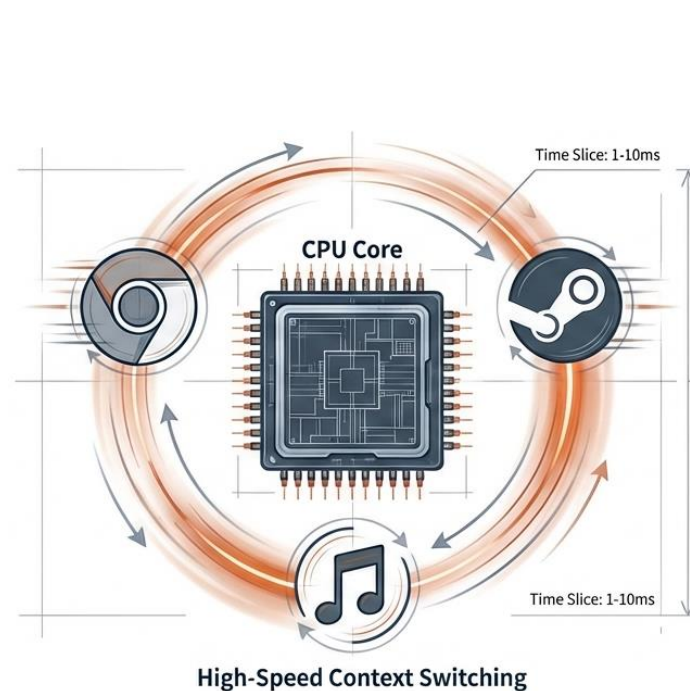
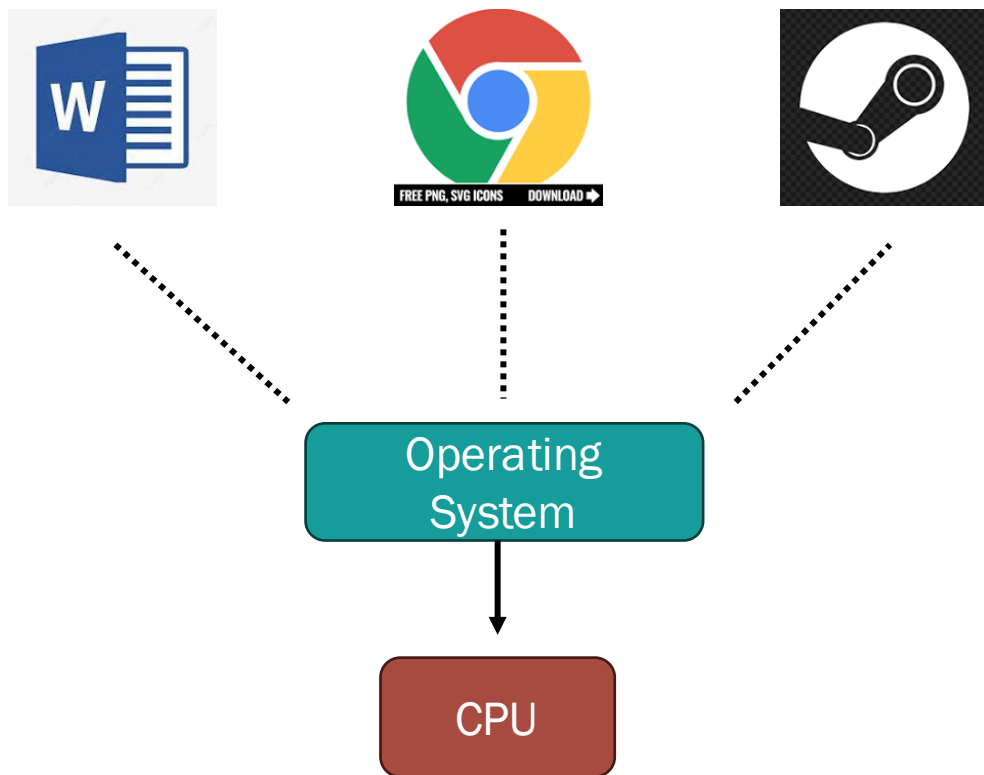
プロセスとは？ - 実行中のプログラムのこと。

プロセスの状態遷移 - 「実行中」「待機中」「準備完了」

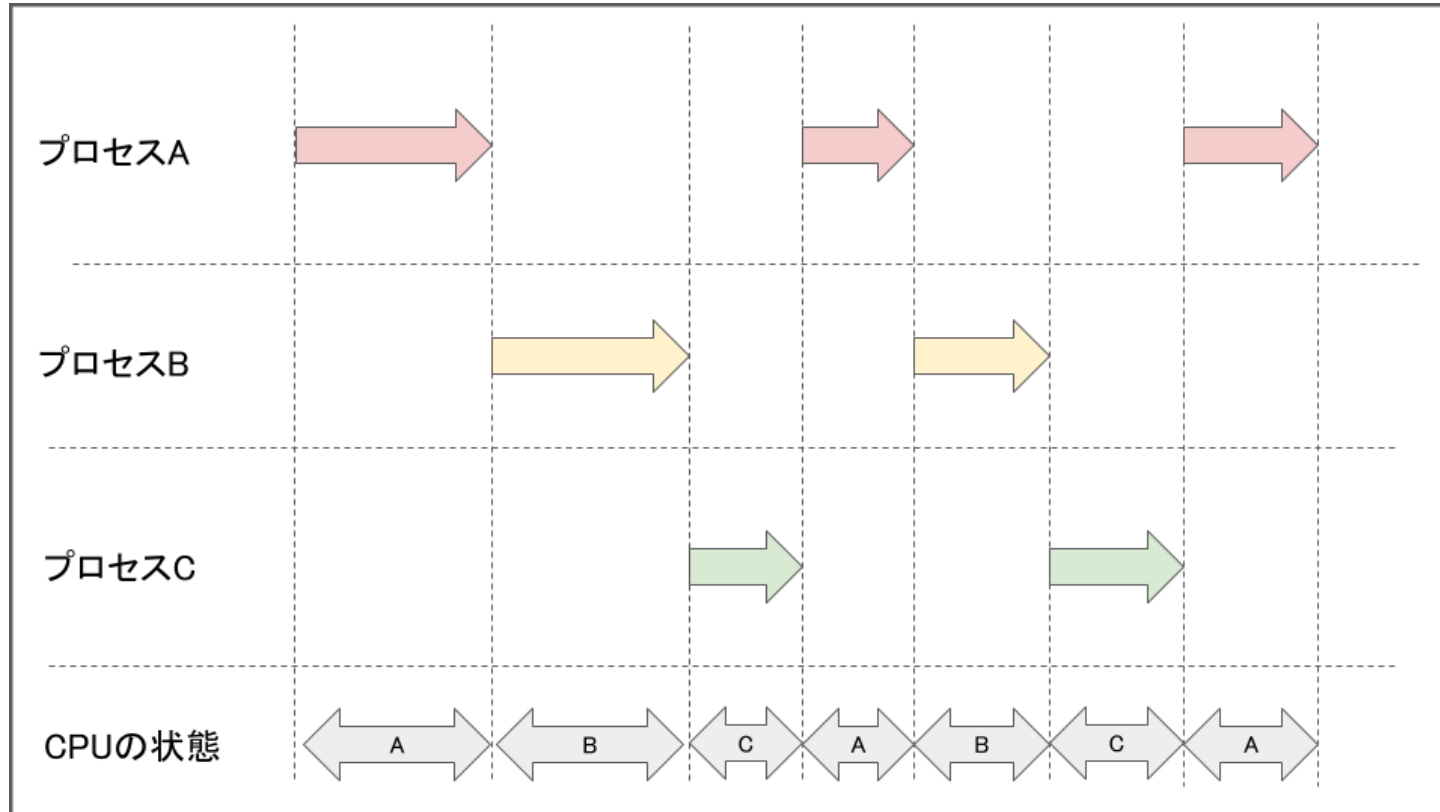


機能① プロセス管理：マルチタスクの実現

この高速な切り替え（タイムシェアリング）により、人間の目には全てのプログラムが「同時に」動いているように見える



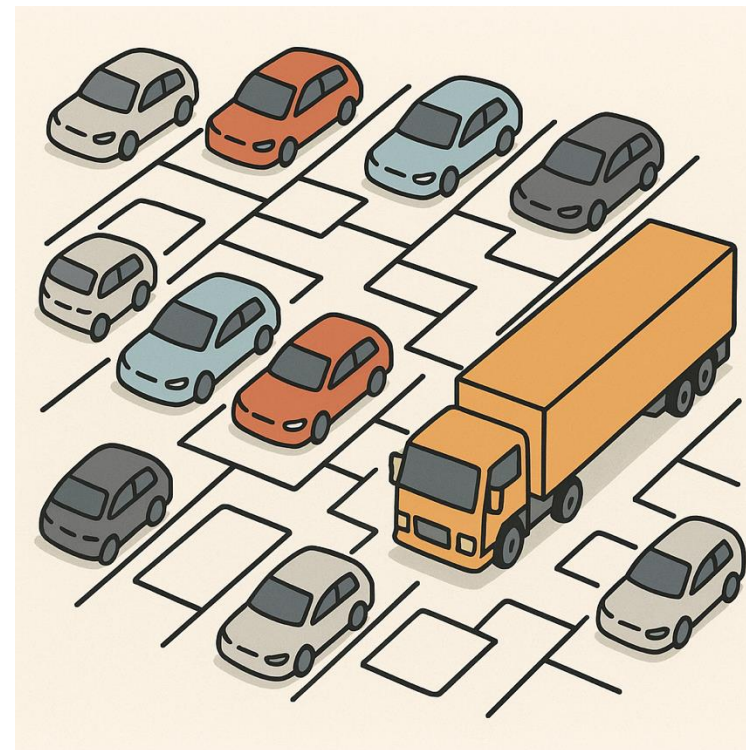
機能① プロセス管理：コンテキストスイッチ



- CPUがプロセスを切り替える際、現在の作業内容（レジスタの値など）を一旦メモリに保存し、次のプロセスの作業内容を読み込む

機能② メモリ管理

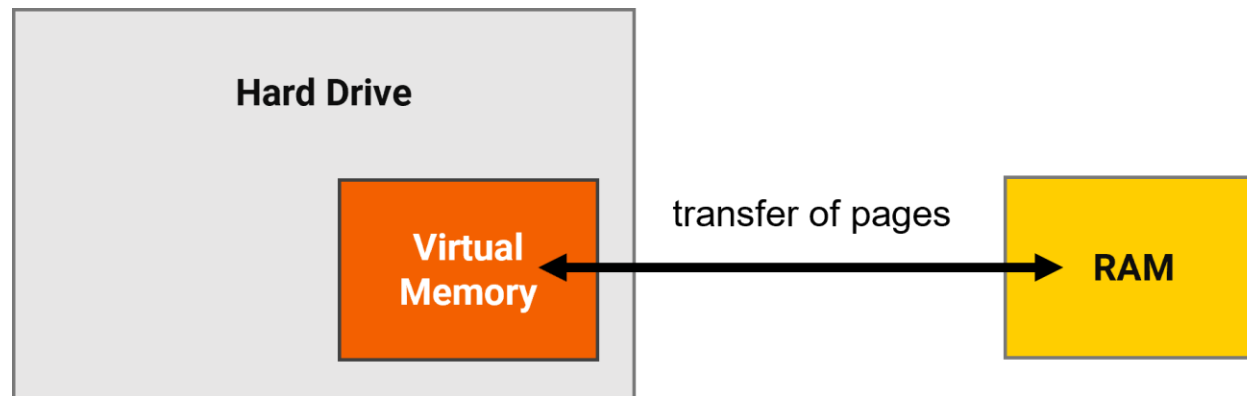
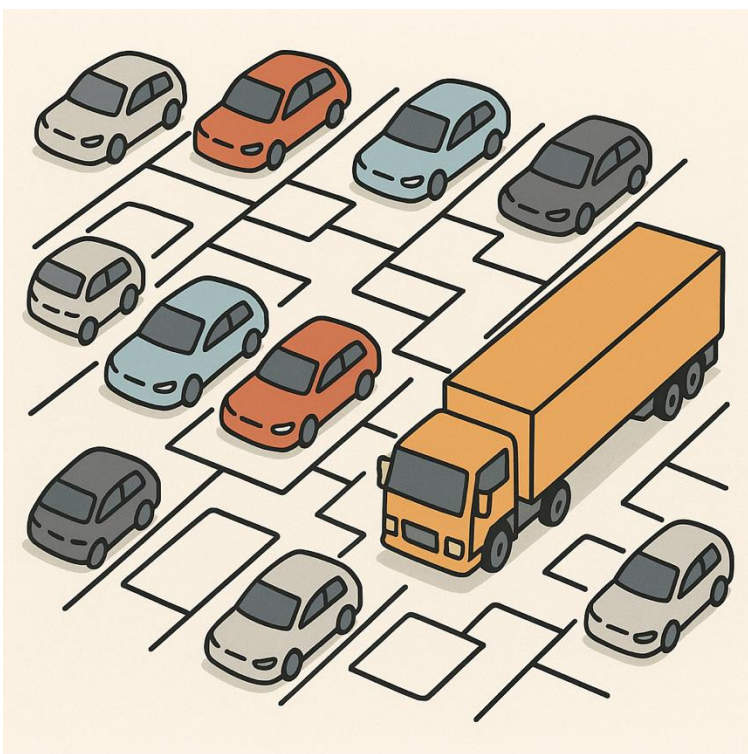
問題点① - あるプログラムが間違っって他のプログラムの領域を書き換えてしまうかもしれない。



問題点② - フラグメンテーション メモリの確保と解放を繰り返すと、細切れの「隙間」だらけになり、大きなプログラムを起動できなくなる

機能② メモリ管理

解決策：仮想メモリ - OSが各プロセスに「あなたは巨大なメモリを丸ごと貸し切りですよ」と見せかける（仮想アドレス空間）



物理的なメインメモリ（RAM）が不足した際に、ハードディスク（HDD）やSSDの一部を一時的な作業領域として利用する仕組み

機能③ ファイルシステム

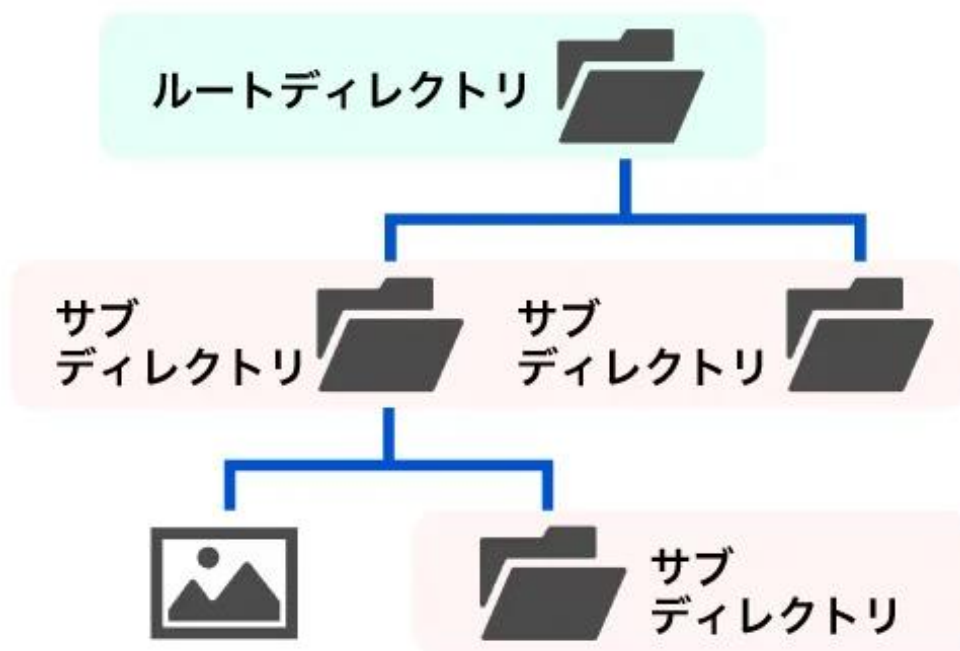
ストレージは、本来ただの膨大な0と1の羅列。
このままではどこに何があるか分からない。



OSが**図書館の司書**のように、膨大な本（データ）を分類し、ラベルを貼り、索引カード（ファイルシステム）を作って管理している。

機能③ ファイルシステム - ファイルとディレクトリ

OSは、データに「ファイル名」を与え、「ディレクトリ（フォルダ）」という階層構造で整理する。



アクセス権管理 - ファイルごとに「読み取り専用」「書き込み可能」といった許可（パーミッション）を設定し、セキュリティを保つ。

抽象化 - 私たちは、物理的なディスクのどこにデータがあるかを一切意識せず、「ファイル名」だけでデータにアクセスできる。

モジュール7：オペレーティングシステム（OS）の役割と仕組み

01

なぜOSが必要か

03

人間との対話方法

02

OSの三大機能

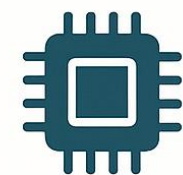
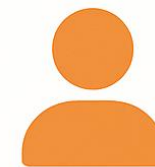
04

OSの種類と歴史

OSのもう一つの顔

OSは、ハードウェアを管理するだけでなく、私たち人間がコンピュータを操作するための「窓口（インターフェース）」も提供する

ハードウェア

人間
(ユーザー)

GUI (Graphical User Interface)



- **GUI (Graphical User Interface)**
- 専門知識がなくても、視覚的・直感的に操作できるインターフェース。

GUIの基本要素

- **アイコン (Icon):** ファイルやアプリを絵で表現。
- **ウィンドウ (Window):** 複数の作業を同時に行うための領域。
- **ポインタ (Pointer):** マウスで直接的に対象を指し示す。



- **GUIの起源**：デスクトップ・メタファー
- **1970年代**：ゼロックス社 パロアルト研究所 (PARC) で誕生。
- **コンセプト**：コンピュータの画面を「本物の机の上」に見立てる。
- 後にAppleのMacintoshによって世界に普及。

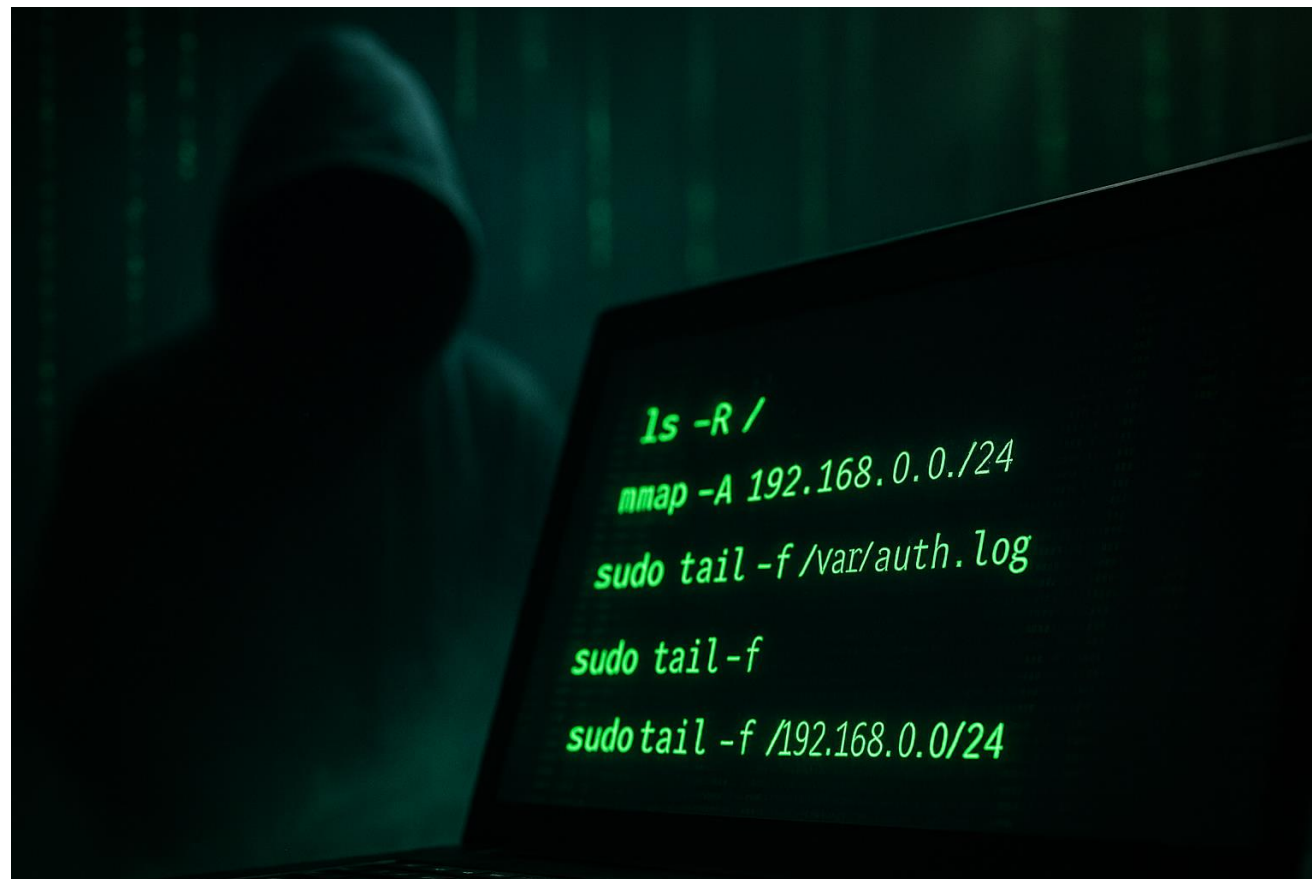


現代GUIの基本設計「WIMP」

- **Window**: 複数の作業領域
- **Icon**: ファイルや機能の視覚的表現
- **Menu**: 選択可能なコマンドの一覧
- **Pointer**: マウスカーソルなどの指示器

もう一つの対話方法 : CLI

- **CLI (Command Line Interface)**
- 映画のハッカーが使っている、あの「黒い画面」の正体
- 文字（コマンド）だけでコンピュータと対話する方法



CLIの仕組み：シェルとカーネル

- **カーネル (Kernel):** OSの心臓部。ハードウェアを直接制御する
- **シェル (Shell):** 人間の言葉（コマンド）をカーネルが理解できる言葉に翻訳する「通訳」



コマンドの基本構造

```
ls -l /Users
```

(コマンド): 何をするか
(例: リスト表示)

(オプション): どうやってするか
(例: 詳細形式で)

(引数): 何を対象にするか
(例: /Usersフォルダを)

`pwd` : Print Working Directory (現在地はどこ?)

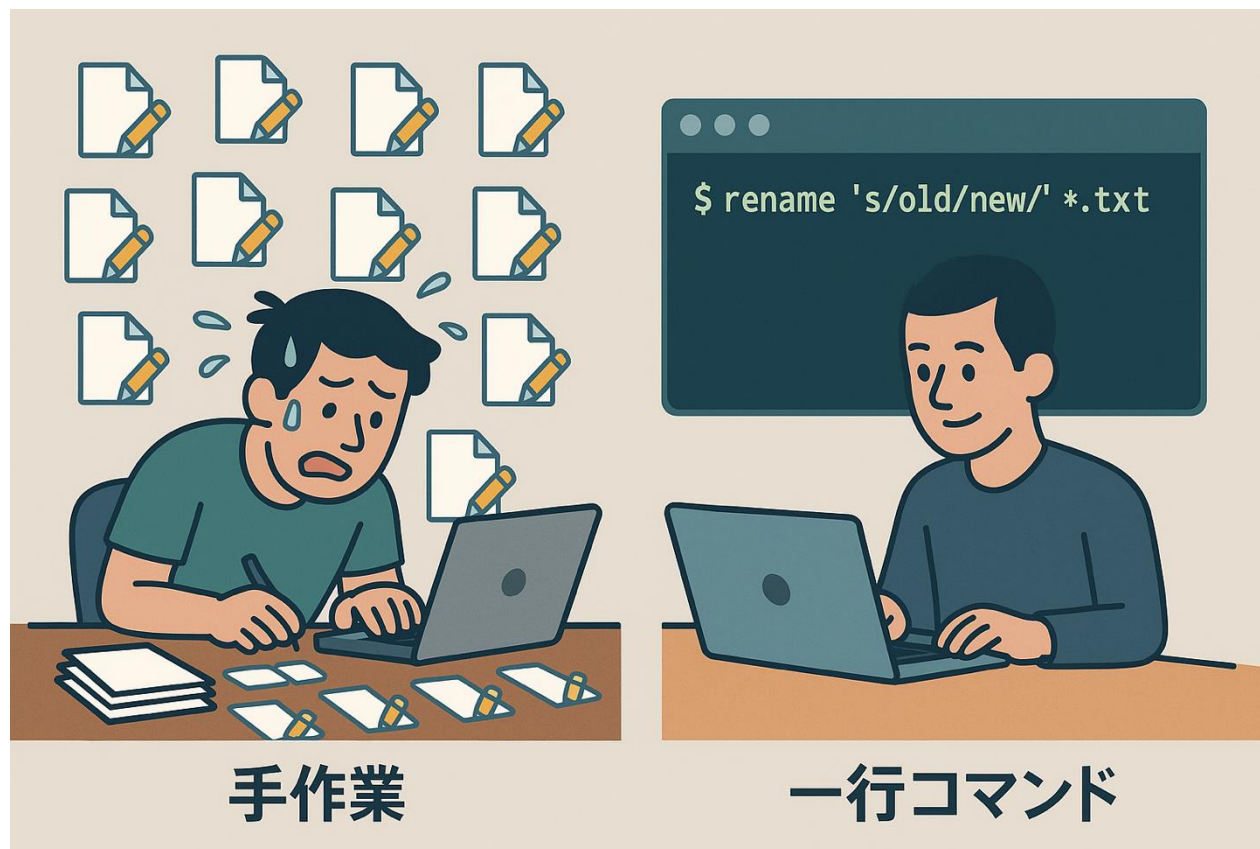
`ls` : List (周りには何がある?)

`cd` : Change Directory (場所に移動する)

`mkdir` : Make Directory (新しい部屋を作る)

なぜ今でもCLIを使うのか？

- 自動化: GUIでは面倒な繰り返し作業を一瞬で完了できる。
- リモート操作: サーバーなど、遠隔地のコンピュータを安全
 - 確実に操作できる。
- 高機能: 複数のコマンドを組み合わせることで、GUIにはない複雑な処理が可能。



GUIとCLIのまとめ



- **GUI:** 直感的、分かりやすい（汎用的）
- **CLI:** 高速、高機能、自動化（専門家向け）

モジュール7：オペレーティングシステム（OS）の役割と仕組み

01

なぜOSが必要か

03

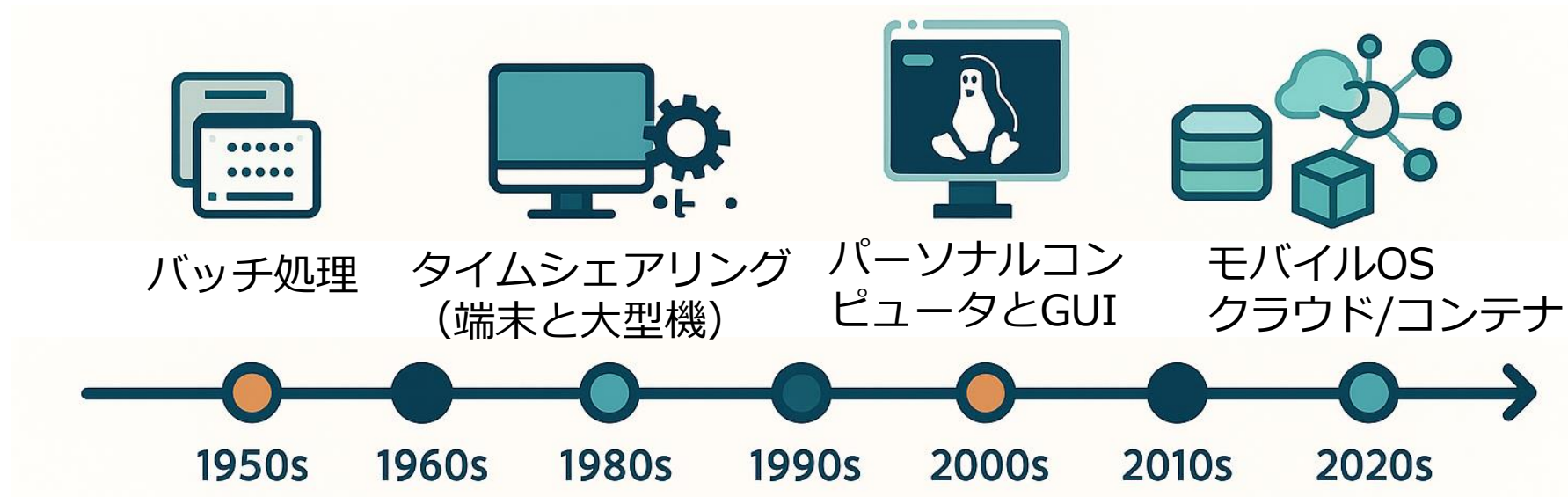
人間との対話方法

02

OSの三大機能

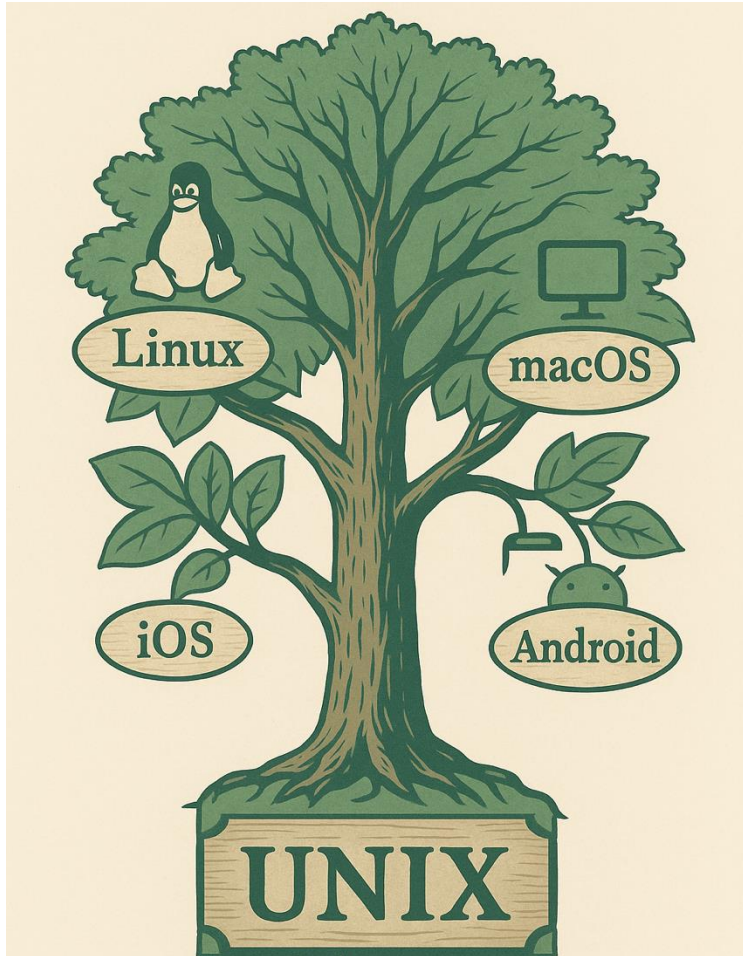
04

OSの種類と歴史



一度に一つの仕事しかできなかった時代から、多くの仕事を同時にこなすのが当たり前の時代へ。

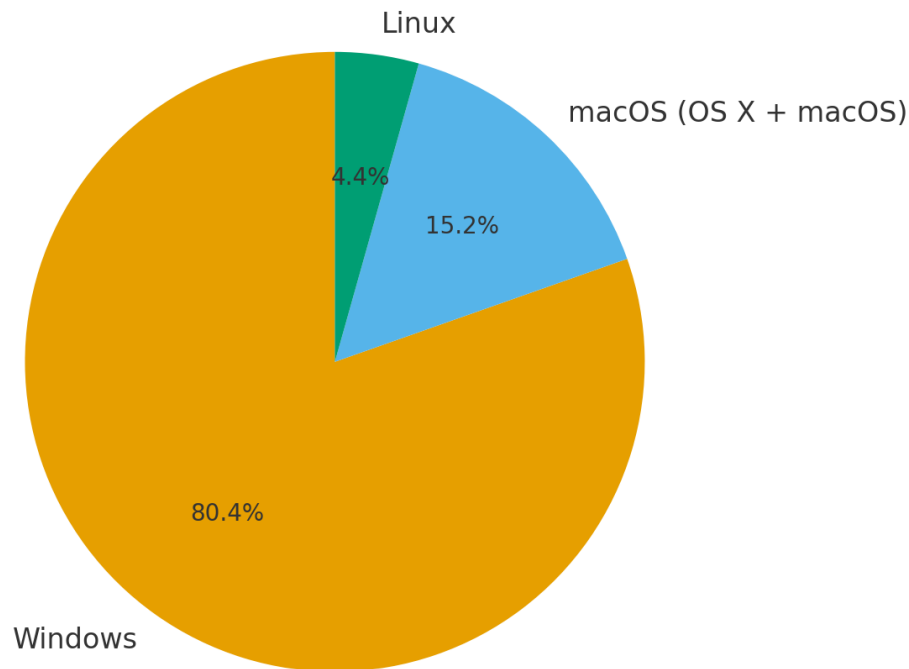
OSの源流 : UNIX



現代の多くのOSは、1969年に生まれたUNIXというOSの思想を色濃く受け継いでいる。

OSの勢力図 (デスクトップ)

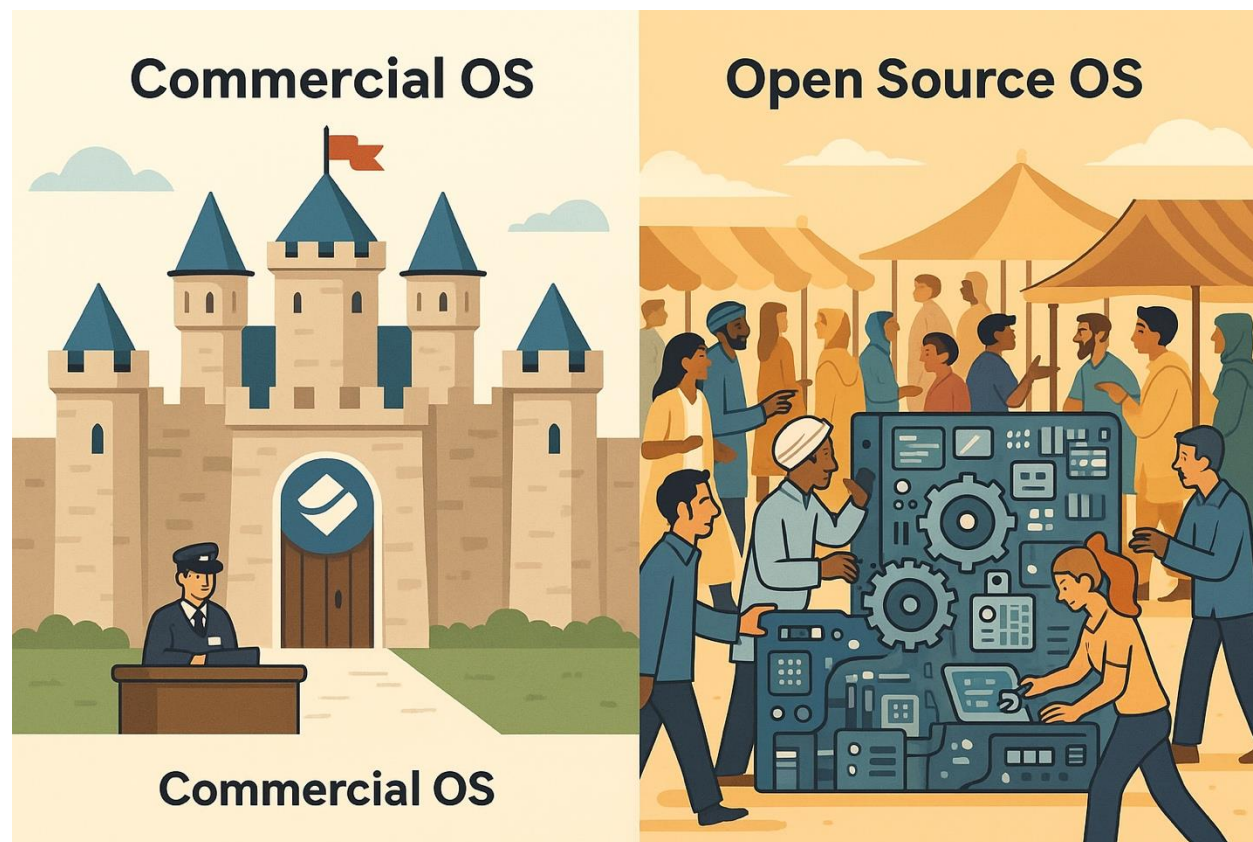
Desktop OS Market Share (Windows vs. macOS vs. Linux)
Worldwide · StatCounter · Jul 2025 (normalized to these three)



- **Windows:** 圧倒的なシェアを誇る巨人。
- **macOS:** UNIXベースのApple製OS。
- **Linux:** UNIXライクなオープンソースOS。サーバーや研究開発で絶大な力を持つ。

開発モデルの違い：商用 vs オープンソース

- **商用OS (Windows, macOS):**
 - 特定の企業が開発・販売。
 - 設計図（ソースコード）は非公開（クローズドソース）。
 - ビジネスとして、利益とサポートを重視。
- **オープンソースOS (Linux):**
 - 世界中のボランティア（コミュニティ）が共同で開発。
 - 設計図（ソースコード）は全世界に公開。
 - 知識の共有と技術の自由な発展を重視する「文化」。



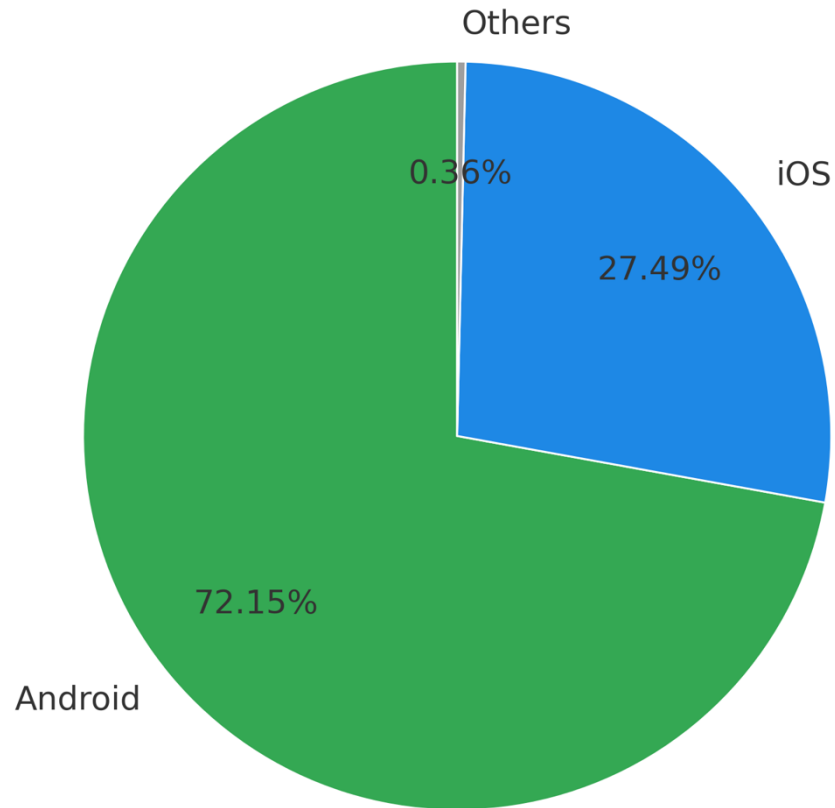
オープンソースという「文化」



- **オープンソースの核心:** ソースコードを公開し、誰でも自由に利用・修正・再配布できる。
- **メリット:**
 - **透明性:** バグやセキュリティの脆弱性が発見されやすい。
 - **革新性:** 世界中の才能が集まり、進化のスピードが速い。
 - **教育:** 最高の教材として、誰もが中身を学べる。
- Linuxは、このオープンソース文化の最も成功した象徴。

OSの勢力図（モバイル）

Mobile OS Market Share — Worldwide (Jul 2025, StatCounter)



スマートフォンの世界は、共にUNIXの思想を受け継ぐこの2つのOSが市場を二分している。



最終回：未来のコンピュータ技術
と応用の展望