

1. イントロダクション：オペレーティングシステム (OS) の本質

- OS とは何か？

- 拡張マシン (Extended Machine) としての OS

- 裸のハードウェア (bare hardware) の複雑さを隠蔽し、より高度な機能を持つ「拡張マシン」(使い勝手の良い計算機)を提供する。
 - この操作は「**抽象化 (abstraction)**」と呼ばれる技術によって実現される。
 - 裸のハードウェア、OS、システムライブラリ、システムユーティリティ、GUIライブラリなどのシステムコンポーネントが「抽象化の層」を成している。

- リソースマネージャ (Resource Manager) としての OS

- コンピュータを構成するプロセッサ、メモリ、タイマ、入出力デバイス (ディスク、マウス、キーボード、ネットワークアクセスと帯域) といったすべての資源を、「ある方針」のもとに、それらの資源を利用して仕事を行おうとしている多くのプログラムに割り当てる。
 - 資源利用の**多重化 (共有)**や**排他制御**を行う。
 - 計算に関わる資源 (時間、空間 (メモリ)、デバイス) の量と使用タイミング (共有、排他的利用) を管理する。

- 抽象化の例：ファイル処理

- ユーザープログラムは、物理的なハードディスク装置の複雑な詳細 (ヘッド番号、シリンダ番号、セクタ番号) を知ることなく、「ファイル名」 (例: "test.txt") という抽象化された概念でデータを読み書きできる。
 - 「ファイル操作」は、OS (ファイルシステム、デバイスドライバ) を介して、ハードウェアの「セクタの読み書き」というI/O命令へと変換される。
 - このプロセスは、ユーザープログラムからOSへの**システムコール**、OSからハードウェアへのI/O命令という「抽象の実現」のレイヤー構造によって行われる。

- OSを構成する主要な概念 (抽象)

- プロセス (Process)
 - メモリ管理 (Memory management)
 - 入出力 (Input / Output)
 - ファイル (File)
 - セキュリティ (Security)
 - (基本的な2つの概念として「プロセス」と「入出力システム」が挙げられることもある。)

- システムコール

- OSが提供する「拡張マシン」の命令に相当する。
 - 特殊なマシン命令 (システム割り込み、割り出し命令、**トラップ命令**) で起動される。
 - 通常のコール命令と異なり、CPUの「**状態遷移 (ユーザーモード → システムモード)**」を伴う。
 - システムコールから見た OS の概念の整理
 - **プロセス管理**: fork, wait, execve, exit など。
 - **ファイル管理**: open, close, read, write, lseek, stat など。
 - **ディレクトリ・ファイルシステム管理**: mkdir, rmdir, link, unlink, mount, umount など。
 - **その他**: chdir, chmod, kill, time など。

- OSの構造

- モノリシック (Monolithic)
 - 階層システム (Layered Systems)
 - 仮想マシン (Virtual Machines)
 - マイクロカーネル (Microkernels) / 外部カーネル (ハイパーバイザ型)
 - クライアント・サーバモデル (Client-Server Model)

- 多様な環境・用途向けのOS

- パソコン用OS、サーバ用OS、マルチプロセッサ用OS、リアルタイム用OS、組み込み用OS、スマートカード用OS、メインフレーム用OSなど。

3. シラバス概要（講義内容）

- 1. イントロダクション・概要
- ファイルシステム
 - 12. ファイル、ディレクトリ
 - 13. ファイルシステムの実装
 - 14. ファイルシステムの例
- プロセス
 - 2. プロセス、スレッド
 - 3. プロセス間通信、同期
 - 4. スケジューリング
- メモリ管理
 - 7. 仮想記憶、ページング、ページテーブル
 - 8. ページ置き換えアルゴリズム
 - 9. 設計時の課題、セグメンテーション
- デッドロック
 - 5. リソース、デッドロック
 - 6. デッドロックの検出、回復、回避、防止
- 入出力
 - 10. 入出力1 (I/O ハードウェア)
 - 11. 入出力2 (I/O ソフトウェア)

重要な用語の抽出とその説明

1. **オペレーティングシステム (OS)**: 裸のハードウェアを隠蔽し、より高度な機能を持つ「拡張マシン」を提供する、またコンピュータのすべての資源をプログラムに割り当てる「リソースマネージャ」の役割を持つソフトウェア。
2. **抽象化 (Abstraction)**: 複雑な詳細を隠蔽し、より簡潔で扱いやすい概念として提供する技術。OSはハードウェアの抽象化を提供する。
3. **拡張マシン (Extended Machine)**: OSがユーザーやアプリケーションに対して、bare hardware（裸のハードウェア）よりも高度で使いやすい機能を提供する仮想的な計算機。
4. **リソースマネージャ (Resource Manager)**: OSが、プロセッサ、メモリ、タイマ、入出力デバイスなどの計算機資源を、複数のプログラムに対して「ある方針」のもとに効率的に割り当てる機能や役割。
5. **システムコール (System Call)**: OSが提供する「拡張マシン」の命令に相当し、特殊なマシン命令（トラップ命令など）によってCPUの状態をユーザーモードからシステムモードに遷移させて実行される。
6. **ファイル (File)**: 物理的な記憶装置の詳細を隠蔽し、ユーザーやプログラムが名前ですべてデータを扱えるようにした抽象化されたデータ単位。
7. **ディレクトリ (Directory)**: ファイルシステム内でファイルを階層的に整理し、名前によって管理するための抽象化された構造。
8. **プロセス (Process)**: OSを構成する主要な抽象概念の一つで、実行中のプログラムを指し、CPUなどの資源が割り当てられ独立して実行される活動の単位。
9. **メモリ管理 (Memory Management)**: OSを構成する主要な抽象概念の一つで、物理メモリ資源を効率的に割り当て、複数のプログラムが安全に利用できるようにするための機能。
10. **入出力 (Input / Output - I/O)**: OSを構成する主要な抽象概念の一つで、コンピュータと外部デバイス（ディスク、マウス、キーボードなど）との間でデータをやり取りする機能。
11. **セキュリティ (Security)**: OSを構成する主要な抽象概念の一つで、システムの安全性やデータの保護に関わる機能。
12. **トラップ命令 (Trap instruction)**: システムコールを起動する特殊なマシン命令で、CPUの状態をユーザーモードからシステムモードへ遷移させる機能を持つ。
13. **CPUの状態遷移 (User mode → System mode)**: システムコール実行時にトラップ命令によって発生するCPU

の動作モードの切り替わりで、OSがハードウェアへのアクセスや機密性の高い処理を行う際に使用される。

14. **コンピュータアーキテクチャ (Computer Architecture):** コンピュータシステムの設計に関する概念で、命令、プログラムカウンタ (PC)、演算機構 (ALU)、メモリサイクル、デバイス、割り込みなどが含まれ、OS学習の前提知識となる。

追加で学習すべき項目

ご提供いただいた資料は、OSの基本的な概念と講義のシラバスを明確に示しています。これらの資料を最大限に活用し、より深い理解を得るために、シラバスに挙げられているものの、本資料では詳細に解説されていない以下の項目を追加で学習することをお勧めします。

1. プロセスとスレッド (Processes & Threads) の詳細

- 資料では「プロセス」がOSの基本的な概念として挙げられていますが、現代のOSにおける**スレッド**の概念、プロセスとの違い、スレッドのメリット（軽量性）、ユーザーレベルスレッドとカーネルレベルスレッドの違いなどについて詳しく学ぶことで、OSにおける並行処理の仕組みを深く理解できます。

2. プロセス間通信 (InterProcess Communication - IPC) と同期

- 複数のプロセスが協調して動作するための、**情報交換**や**同期**のメカニズムは非常に重要です。**パイプ、メッセージキュー、共有メモリ、セマフォ、ミューテックス、モニタ**などの具体的なIPC機構とその実装、競合状態の回避策について学ぶことが推奨されます。

3. スケジューリング (Scheduling)

- CPUという限られたリソースを複数のプロセスやスレッドがどのように共有し、どの順番で実行されるかを決定するOSの機能です。**FCFS (First-Come, First-Served)、SJF (Shortest Job First)、優先度ベース、ラウンドロビン**など、さまざまなスケジューリングアルゴリズムとその特性、またそれらがシステム性能に与える影響について学ぶことが重要です。

4. 仮想記憶、ページング、ページテーブル、ページ置き換えアルゴリズム、セグメンテーション

- メモリ管理の章に詳細な項目が並んでいます。**仮想アドレス空間、物理アドレス空間、アドレス変換機構、ページングとページテーブルの仕組み、ページフォールト処理、そしてNRU、FIFO、LRU、ワーキングセット**などのページ置き換えアルゴリズムについて深く学ぶことで、メモリ資源の効率的な利用と保護の仕組みを理解できます。**セグメンテーション**についても理解を深めると、メモリ管理の選択肢が広がります。

5. デッドロック (Deadlocks) の検出、回復、回避、防止

- 複数のプロセスが互いにリソースの解放を待ち合い、システムが停止する**デッドロック**はOSが解決すべき重要な問題です。デッドロック発生の**4つの必要条件**（相互排他、保持と待機、非プリエンプション、循環待機）を理解し、**デッドロックの検出アルゴリズム、回復戦略、回避アルゴリズム**（例: 銀行家のアルゴリズム）、および**防止策**について学ぶことは、信頼性の高いシステム設計に不可欠です。

6. 入出力ハードウェアとソフトウェア (I/O hardware & software)

- I/Oデバイスの特性、デバイスコントローラ、**デバイスドライバ**の役割、I/O処理の仕組み（**ポーリング、割り込み駆動I/O、DMA (Direct Memory Access)** など）について学ぶことで、OSが物理的なデバイスとどのように連携し、効率的なデータ転送を実現しているかを理解できます。

7. OSの構造 (Operating System structure) の詳細

- モノリシック、階層システム、仮想マシン、マイクロカーネル、クライアント・サーバモデルなど、さまざまなOS構造が列举されています。それぞれの構造の**特徴、利点、欠点**、そして現代のOSがどのモデルを採用しているかを学ぶことで、OS設計思想の多様性を理解できます。

8. コンピュータアーキテクチャの前提知識の補強

- 「マシン語に関する若干の知識」や「コンピュータアーキテクチャ（命令、プログラムカウンタ、演算機構、メモリサイクル、デバイス、**割り込み**）」が前提知識として挙げられています。特に**割り込み**はシステムコールやI/O処理の根幹に関わるため、そのメカニズムを詳しく学ぶことは、OSの動作原理を深く理解する上で非常に重要です。

これらの項目は、提供された資料のシラバスに明記されている講義内容であり、OS全体の包括的な理解を深めるために大変有効です。

