

# 第1章 オペレーティング システムの役割

花田 英輔  
(このPowerPointは渡辺名誉教授作成のものを花田が一部改編した)

1

## オペレーティングシステムとは(教科書1.1)

応用プログラム=計算資源の要求

OS=両者を調整、制御

ハードウェア=計算資源の提供

例えば、OSはグラウンド管理人

グラウンドを使いたい人

管理人=OS (小規模ならなくても)

グラウンド

調整・制御  
場所分割  
時間分割  
環境整備

2

## OSの役割(教科書1.2)

キーボード  
命令受付  
プログラム動作制御  
OS  
保護  
入出力代行  
応用プログラム  
モニター  
プリンタ  
ディスク  
OSは様々な業務をこなしている

3

## OSが提供する機能(教科書1.3)

利用者 → 応用プログラム

ユーザインターフェース (ユーザが直接操作)

(アプリケーション)プログラムインターフェース(API) (ユーザは間接的に操作)

OS

ハードウェア

2つのインターフェースの区別は重要

4

## ユーザインターフェース (UI)

- ▶ 「インターフェース (Interface) = システム間又はシステムと人間の接点となる部分」
- ▶ ユーザインターフェース = ユーザが直接にOSに対してコマンド等で指示を出すインターフェース
  - GUI (Graphical User Interface) = 操作をアイコン、マウス等で行えるようにしたもの
  - CUI (Character-based User Interface) = 文字表示、文字入力による操作。コマンドラインインターフェース (Command Line Interface) とも言う。

5

## プログラミングインターフェース

- ▶ プログラミングインターフェース = プログラムからOSを操作するインターフェース
- ▶ アプリケーションプログラミングインターフェース (API) とも言う

open()  
read()  
write()  
close()  
exit()  
.....

OS

プログラム

6

### OSが管理する資源(教科書1.4)

▶ **資源** (Resource、リソース)

- 作業に必要なモノ
- ハードウェア資源: プロセッサ、メモリ、外部記憶、プリンタ等
- ソフトウェア資源: プログラムやデータ(ファイルとして管理)

▶ OSは資源を応用プログラムに矛盾なく使わせる

The diagram illustrates the OS's role in resource management. A user (利用者) interacts with application programs (応用プログラム) and application programs (応用プログラム). The OS (OS) manages software resources (ソフトウェア資源の管理) and hardware resources (ハードウェア資源の管理). Hardware resources include the processor (プロセッサ), memory (メモリ), and printer (プリンタ). Software resources include files (ファイル) and disks (ディスク).

7

### OSの利用形態(教科書1.5)

▶ コンピュータの利用形態は、用途や時代により様々

- バッチ処理(一括処理)
- オンライン・トランザクション処理
- 時分割処理
- 実時間処理
- 個人使用⇒パソコン(PC、Personal Computer)
- ネットワーク連携

8

### OS以前の処理

- 機械語記述**プログラム  
⇒スイッチを操作して入力、スタートボタンで開始

The diagram shows a person (利用者) at a computer terminal (コンピュータ) setting up their own program (自分のプログラムを設定して開始).

- ハードウェア実験のような形態
- 高水準言語記述**プログラムの出現  
⇒操作が非常に面倒に

9

### 高水準言語プログラムの処理

- ▶ **コンパイラ**: Compiler
  - ソースプログラムを対応するオブジェクトプログラムに変換する(コンパイルする)プログラム
- ▶ **リンカ**: Linker
  - オブジェクトプログラムに関連ライブラリを加えて、実行可能プログラムにする(リンクする)プログラム
- ▶ **ローダ**: loader
  - プログラムをメモリ上に展開する(ロードする)プログラム

The flowchart shows the process: Source Program (ソースプログラム) is compiled by a Compiler (コンパイラ) into an Object Program (オブジェクトプログラム). The Object Program is then linked with libraries (ライブラリ) by a Linker (リンカ) to create an Executable Program (実行可能プログラム). Finally, the Executable Program is loaded (ロード) into memory by a Loader (ローダ) for execution (実行), which produces output data (出力データ) from input data (入力データ).

10

### 高水準言語プログラムの処理 (その1 コンパイル)

The diagram shows the compilation process: A user (利用者) sets the loader (ローダを設定) and starts the compiler (コンパイラ) at the start position (開始位置を設定し、ローダをスタート). The compiler (コンパイラ) reads the source program (読み込みプログラム) and writes out the object program (書出し、オブジェクトプログラム).

11

### 高水準言語プログラムの処理 (その2 リンク)

The diagram shows the linking process: A user (利用者) sets the loader (ローダを設定) and starts the linker (コンパイラ) at the start position (開始位置を設定し、ローダをスタート). The linker (コンパイラ) reads the object program (読み込み、オブジェクトプログラム) and links it with libraries (ライブラリ) to create an executable program (書出し、実行可能プログラム).

12

### 高水準言語プログラムの処理 (その3 実行:OS以前の処理)

処理途中でおかしくなったら、また最初からやりなおし

利用者にとって面倒・難解⇒専門の操作員へ  
⇒専門の操作員も面倒・難解⇒できるだけ自動化⇒初期のOSへ

13

### OSの利用形態(教科書1.5)

- ▶ コンピュータの利用形態は、用途や時代により様々
  - バッチ処理(一括処理)
  - オンライン・トランザクション処理
  - 時分割処理(Time Sharing)
  - 実時間処理(Real Time)
  - 個人使用⇒パソコン(PC、Personal Computer)
  - ネットワーク連携

14

### 運用形態

- ▶ 時間借り
  - 時間が未確定(1時間予約でも実処理は10分?2時間?)
  - 操作が面倒(コンパイル⇒リンク⇒実行をロード・実行・監視)
- ▶ 操作員に依頼
  - 利用者は操作員にプログラム、データ、指示書を渡す
  - 操作員はプログラムを投入。**停止を判断して次の処理投入**

15

### バッチ処理(一括処理)

- ▶ 順次実行を自動化
  - ジョブ=手順指示&プログラム&データの一式
  - 操作員は、ジョブをシステムへ投入
  - 利用者は、後日に結果を受け取り

16

### バッチ処理の効率的処理

- ▶ 入力装置・出力装置の速度が遅い
  - 高速なディスクへ溜めて処理
- ▶ SPOOL(Simultaneous Peripheral Operation On-Line)

17

### スプール

- ▶ SPOOL 糸巻き
- ▶ Simultaneous Peripheral Operation OnLine
  - 同時の 周辺の 操作 オンライン

- 入出力処理は計算処理より遅いので、処理の実行と並行実行させて効率向上
- そのためのディスク領域=スプール領域 (印刷スプール領域など現在も)

18

### オンライン・トランザクション処理

- ▶ 多数端末からのデータ検索・更新など **一過性**の処理
  - 「要求⇒応答」で処理完了
  - 座席予約・銀行オンライン等

19

### 時分割処理 (TSS、Time Sharing System)

- ▶ 開発などでの長時間利用
- ▶ 各利用者のプログラムを **短い時間で切り替え実行**
- ▶ 複数の利用者が共用、それぞれ占有利用に感じる

OSが各利用者のジョブを短時間(数十ms)で切替えて実行

20

### 実時間処理 (Real Time System)

- ▶ 自動制御等、時間的制約の厳しい分野での利用
  - 工場の生産ライン、航空機・自動車の制御など
  - 例: 0.1秒ごとに計測、それに基づき一定速度に制御など

21

### 個人使用システム

- ▶ **パーソナルコンピュータ**
  - パソコン、Personal Computer, PC
- ▶ コンピュータが安価に⇒個人占有使用へ
- ▶ OSへの要求も大きく変化
- ▶ 目標
  - 多人数が効率良く使える
  - ⇒個人がやさしく使える

22

### ネットワーク連携

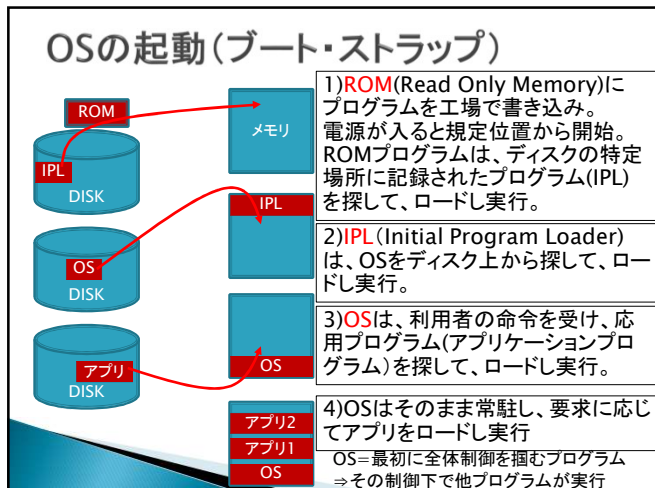
- ▶ インターネットの普及
- ▶ PCは計算処理装置からネットワークの窓口装置へ
- ▶ **クラウドコンピューティング**
  - 実体は雲のようにあいま
  - 世界のどこにあるかわからない?
  - その中で計算処理

23

### コンピュータ起動時のOS

- ▶ 電源投入時のOS起動方法は？
  - メモリ(メインメモリ)は電源が消えれば記憶も消える
  - 外部メモリは電源が消えても記憶は残る
  - 補助記憶(ディスク)とのやりとりはメモリより遅い
- ▶ OSは **2段階**で起動される
  - OSを持ってくる小さなプログラムを呼出す機構を使う
- ▶ 応用プログラムはOSが起動した後に動作可

24

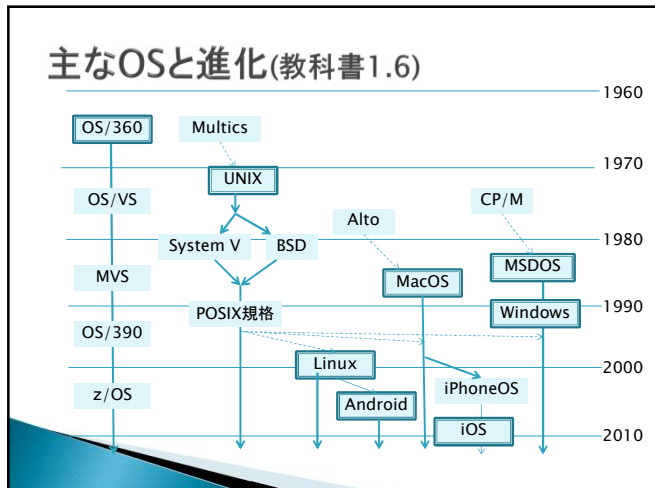


25

### ブート・ストラップ

- ▶ ブート・ストラップ(Bootstrap)
  - ブーツを履くときにブーツを引上げるつまみ革
- ▶ 「独力でやる」という派生意味から、電源投入時に自動的にOSを読み込む方式等に使う

26



27

### パソコン向けOSの発展

- ▶ MacOS=Apple社のMacintosh用OS、最初からGUI、MacOS Xで内部をUNIX系OSに衣替え
- ▶ Windows=マイクロソフト社のPC用OS、CUI基本のMSDOSの拡張としてGUIを実装、その後、新規開発のWindowsNT系へ移行

- ▶ Linux=1991年にスウェーデンの学生によって開発されたUNIX仕様のPC用OS
  - インターネットによる共同開発で発展普及

28

### 今回の課題

1. オペレーティングシステムとハードウェア、アプリケーションプログラムの関係を図示せよ
2. コンピュータの利用形態のうち、バッチ処理、オンラインランザクション処理、時分割処理についてその概要を説明せよ
3. (予習)パソコンのAdministrator(アドミニストレータ)とは何かを調べて記せ

- ▶ 今回のファイル名は“学籍番号-OS02.docx” (例: 22238000-OS02.docx)としてください
- ▶ 締切: 10月20日(金) 18:00 (遅れた場合は減点)

29

### 課題についての注意事項

- ▶ レポートはWordでA4 2ページ程度(表紙含まず)にまとめること
  - 提出はeラーニングシステムを通じて行うこと
- ▶ 締切はその週の金曜日18:00
  - 提出が遅れた場合は減点
  - 大幅な場合はさらに減点の可能性有

#### 記載時の注意事項

- ▶ 参考資料(Webページ)がある場合は**出典を書くこと**
  - **出典を書かずに引用した場合は減点対象**です

30

**講義に関する注意事項(再掲)**

- ▶ 講義に関する連絡はLive Campusメールで行います
- ▶ 講義の課題はeラーニング経由で課題を提出してもらいますので登録してください
  - 全学eラーニングのMoodleを使用します
    - ・コース名:オペレーティングシステム/後/花田英輔
    - ・登録キー:OS2023
- ▶ 本講義に関する情報は次のWebpageにも掲載します  
<http://www.ai.is.saga-u.ac.jp/~hanada/OS/>

2018/10/4