

# オペレーティングシステム(OS)

柴山 潔

## 2. OSの役割

- OSの発展史
- OSの実際的役割

# コンピュータの世代を特徴付ける指標としてのOS

- **ハードウェア**機構の基盤技術

- (A) **論理素子**

- 情報を**処理**する機構の最小単位

- (B) **メモリ素子**

- 情報を**保持/格納**する機構の最小単位

- **ソフトウェア**機能の基盤技術

- (C) **プログラミング言語**

- 種々の**ソフトウェア**を**開発**する際に使用する人工言語

- (D) **オペレーティングシステム(OS)**

- **ソフトウェア**を**ハードウェア**機構上で**実行**する際に必要な**システムプログラム**

# コンピュータシステムの世代と システムプログラムの世代(概略)

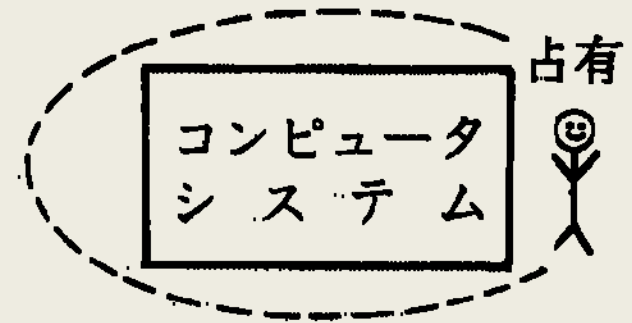
コンピュータの世代(俗称;年代): **システムプログラムの世代**(俗称)

- (1) 第1世代(真空管時代;1940年代~1950年代): **システムプログラムがない**世代
- (2) 第2世代(トランジスタ時代;1950年代~1960年代): **簡易システムプログラム**世代
- (3) 第3世代(IC時代;1960年代~1970年代): **本格的システムプログラム**世代
- (4) 第4世代(マイクロプロセッサ時代;1970年代~1980年代): **統合プログラミング環境**世代
- (5) 第5世代(インターネット時代;1980年代~1990年代): **隠ぺいされたシステムプログラム**世代
- (6) 第6世代(ユビキタス時代;1990年代~2010年代): **透明なシステムプログラム**世代

# 第1世代コンピュータシステムの特徴

## 真空管時代

- 論理素子に真空管
- システムプログラムなし



プログラム  
ハードウェア

マシ語

## 第1世代のOS —システムプログラムがない世代—

- 一人のユーザや単一のプログラムがシステムを占有
- 実行時にハードウェアやユーザプログラムを管理する必要なし

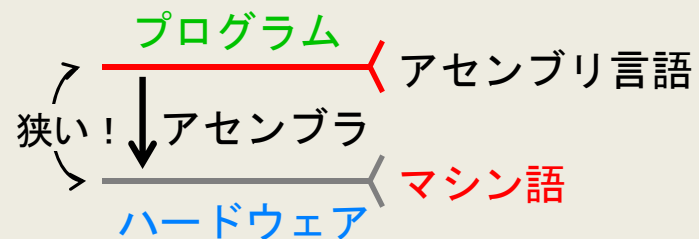
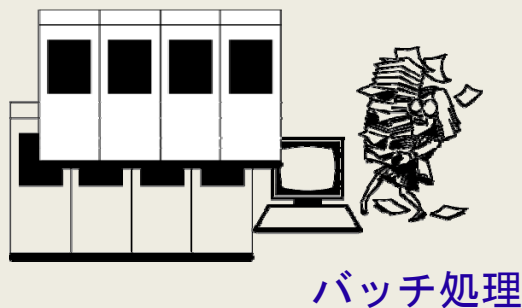
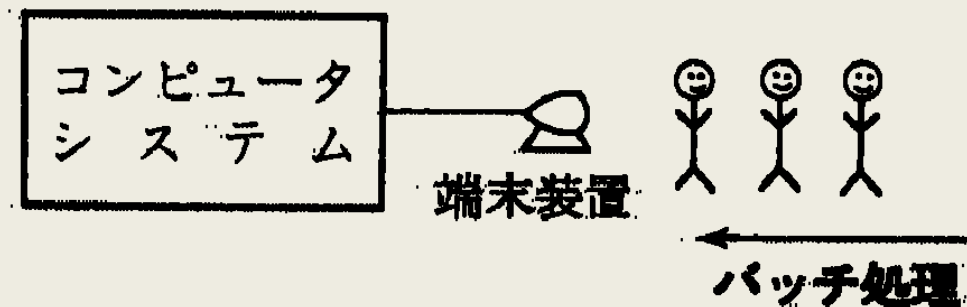


OS は不要！

# 第2世代コンピュータシステムの特徴

## トランジスタ時代

- 論理素子にトランジスタ
- メモリ素子は磁気コアが主
- 簡易システムプログラム
- 高級プログラミング言語 (Fortran, Algol, Cobol)の開発



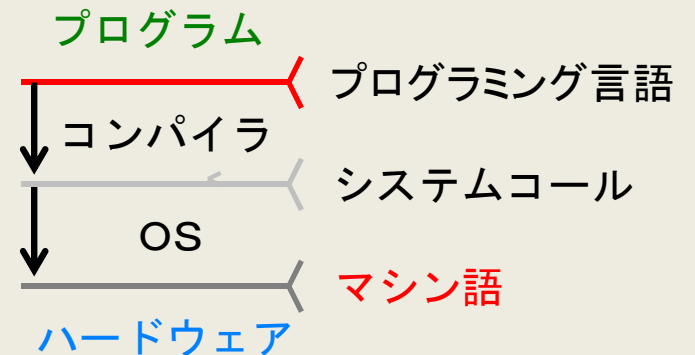
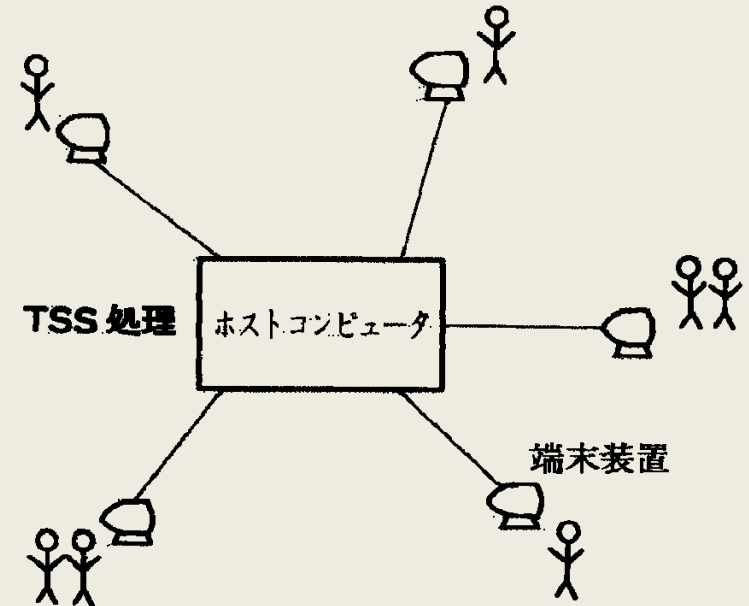
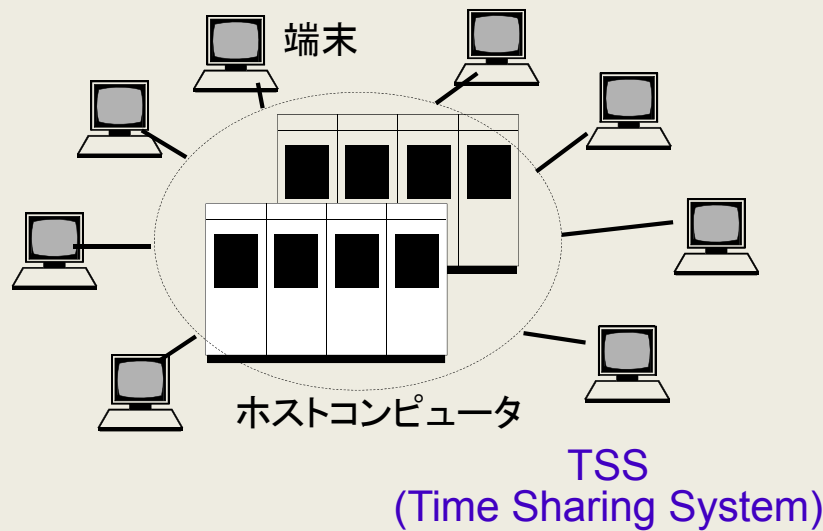
## 第2世代のOS —簡易システムプログラム世代—

- バッチ (batch; 一括) 処理専用のOS (=バッチ型OS) が主
  - バッチ処理: 複数人のユーザや複数個のユーザプログラムを一括して管理して処理/実行
  - 簡素な割り込み処理プログラムだけ

# 第3世代コンピュータシステムの特徴

IC時代

- 論理素子がIC(集積回路)
- メモリ素子も半導体(IC)メモリが主流
- **本格的システムプログラム**





## 第3世代のOS (1)—本格的システムプログラム世代—

- 代表的なシステムプログラムとして、実行前機能のコンパイラと実行時機能のOSとが明確に役割分担
  - バッチ処理指向のOS
- TSS処理用OS = 会話型OS
  - 集中処理 = 会話型処理: 1台のホストコンピュータ (host computer; 処理の中心となる大型コンピュータ) が多数の遠隔端末装置からの要求を集中的に順次処理
  - TSS処理: ホストコンピュータによる計算/処理の時間をミリ秒単位に分割, 順次複数の端末装置 (ユーザ) に割り当て = 時分割利用
    - 複数ユーザがホストコンピュータを同時・多重で利用に見せかけ
  - 会話型システム: ユーザ (プログラム) とシステム (プログラム, OS) とが相互に会話/対話しながら処理を遂行

(上級)

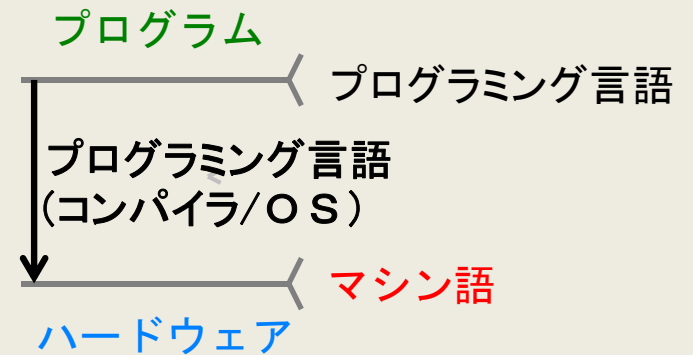
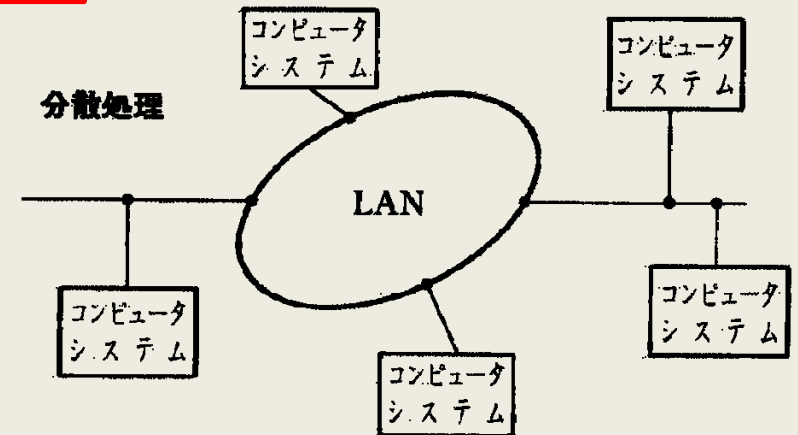
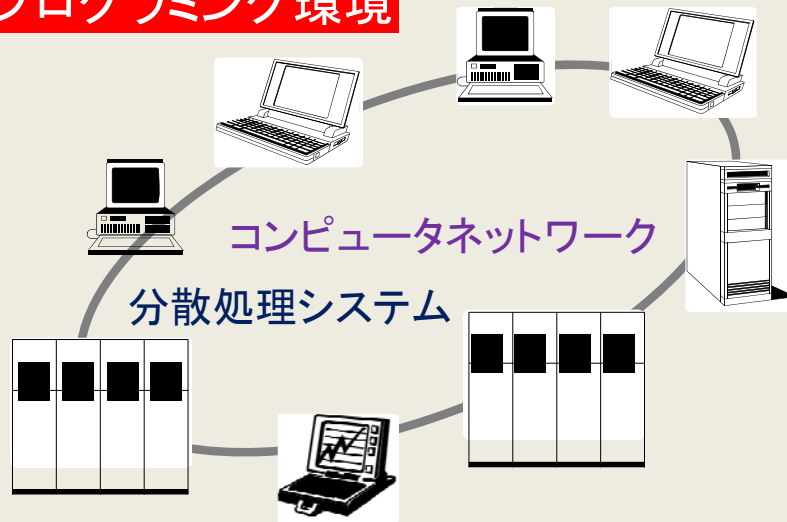
## 第3世代のOS (2) —OS関連のアーキテクチャ技術—

- マルチタスキング(multi-tasking): 資源の多重化技術とその管理・制御技術
- 仮想メモリ: メインメモリの高機能化技術
- 互換性(compatibility): 商用のメインフレームコンピュータのシリーズ化およびファミリ化で必須

# 第4世代コンピュータシステムの特徴

## マイクロプロセッサ時代

- マイクロプロセッサの開発
  - ダウンサイジング
- パソコンやワークステーションの出現
- コンピュータネットワークによるコンピュータ相互の常時接続
- **統合プログラミング環境**



## 第4世代のOS —統合プログラミング環境世代—

➤ パソコン用OS, ワークステーション用OS (代表例: UNIX)

= 会話型OS

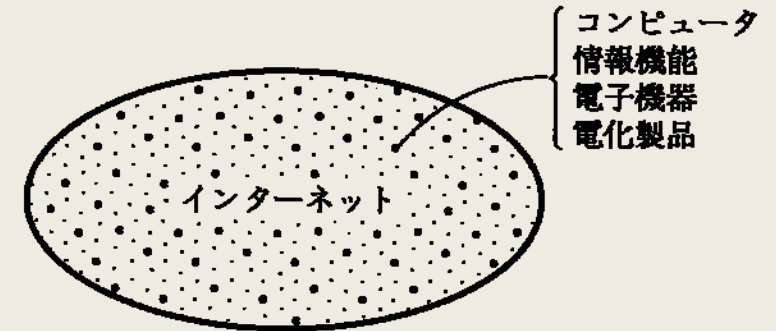
### [特徴]

- 情報を伝達する媒体(メディア)の多種多様化(=マルチメディア(multimedia))に対応・活用するユーザインタフェース(UI)が必須
- LAN (Local Area Network)によって相互結合した複数台のワークステーションが機能分担・協調する分散処理(=分散処理システム)に対応可
- **統合プログラミング環境**: ソフトウェアの多様化, 大規模化および高機能化に対応して, プログラムの開発環境と実行環境の両機能をOSが統合して管理し, ユーザに提供
- 仮想メモリ: メモリ管理機能やファイル管理機能において, メモリ階層の多層化に対応

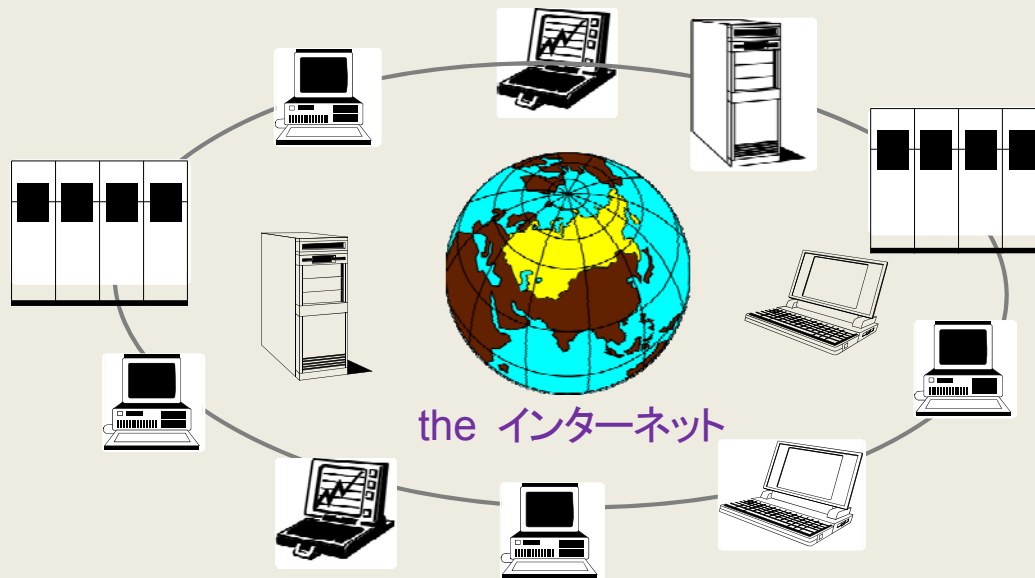
# 第5世代コンピュータシステムの特徴

## インターネット時代

- インターネットの出現
- **隠ぺいされたシステムプログラム**  
← 高度なUIによるOS機能の隠ぺい



ユビキタスコンピューティング



ユーザ

—— ユーザインタフェース

プログラム

—— プログラミング言語

プログラミング環境  
(システムプログラム)

—— (マシン語)

ハードウェア

# 第5世代のOS (1) —隠ぺいされたシステムプログラム世代—

- **UNIX**: ワークステーション用**OS** , **Windows**: パソコン用**OS**

- 第4世代に誕生 → 第5世代に爆発的に普及

## [特徴]

- **インターネットへの接続機能**を標準装備

- 世界中のLANやコンピュータシステムどうしを**常時**接続

- ワールドワイドウェブ(World Wide Web; **WWW**, 単に**Web**ともいう)技術によって世界中に散らばる種々のマルチメディア情報を取得/発信/検索

- 高機能・簡便なユーザインタフェース(=**GraphicalUI**)をユーザに提供

- システムプログラムとしての**OS機能**(=コンピュータシステムの管理やソフトウェアの開発や保守)を**OS自身**でユーザから隠ぺい

- 特に, 情報媒体の**マルチメディア化**は著しく, 画像(動画画像や映像も含む)や音声などをUIの構築に活用

## 第5世代のOS (2) —UNIXの特徴—

➤ **UNIX**: 現代に至る代表的な**会話型OS**

[特徴]

- **グラフィカルUI(GUI)**を装備
  - 高精細グラフィックディスプレイ上で、複数ウィンドウを開け、**アイコンメニューウィンドウ**で操作機能を選択、**マウス**でポインティング指定
- **オープンシステム** (open system; **公開システム**)
  - **OS(UNIX)**を“**C**”というプログラミング言語で記述、そのそのソースプログラムや仕様を積極的に公開
- **分散処理システム** (= **サーバークライアントシステム**) に対応
  - **LAN**で相互接続した複数のワークステーション群を、計算や処理、ファイル管理、通信、入出力などの機能(サービス)ごとに、それらを専用に提供する**サーバ**(server)とそれらのサーバからのサービスを受ける**クライアント**(client)とで役割分担

(上級)

## 第5世代のOS (3) —コンピュータシステム技術—

### [特徴]

- **ハードディスク装置**: 主要な大容量ファイル装置
- **ファイル装置**: フロッピーディスク, CD-ROM, 光磁気ディスク(MO)装置などの多様化
- **PCカード**: フラッシュメモリ/ハードディスクなどの**メモリ系** & モデム/SCSI/ネットワーク接続などの**入出カインタフェース系**の出現・多様化
- **入出カインタフェース**: コンピュータ本体と周辺装置との接続を**無線**(wireless; ワイヤレス)化  
→ **無線ネットワーク**(←第6世代に普及)の端緒



**OS機能に影響**



# 第6世代コンピュータシステムの特徴

- コンピュータネットワークの進化
- ブロードバンド&無線ネットワーク
  - マルチメディア配信
- ユビキタス (*ubiquitous*; いつでもどこでも) コンピュータシステム
  - 無線LAN
- **透明なシステムプログラム**
  - [透明(*transparent*)] = コンピュータシステムやコンピュータネットワーク (ハードウェアおよびその上で動くソフトウェア)をユーザに意識させない



## 第6世代のOS (1) —透明なシステムプログラム世代—

- ユビキタスコンピューティング(ubiquitous computing): コンピュータやインターネットをいつでもどこでも利用可能



### [特徴]

- インターネット接続用の公衆回線のブロードバンド化
- 携帯電話や無線ネットワークの利用によるモバイル(mobile; 機動性のある)ユーザの激増
- インターネットへ簡便に接続できる情報通信機器(代表例: 携帯電話やPDA (Personal Digital Assistant))の普及

(上級)

## 第6世代のOS (2) —コンピュータシステム技術—

### [特徴]

- **TCP/IP**: インターネット経由での**事実上の標準**の接続プロトコル
- **情報セキュリティ**: インターネットやWWWの**公衆化**や**無線化**への対応
- **マルチスレッドアーキテクチャ**: 「動的な実行単位としてのマシン命令列」である**スレッド**
- **WWW**: インターネット上の膨大な情報からの必要な情報の発掘, 新しい情報の創成, さらには, 音楽や映像のリアルタイム配信, 電子商取引, 仮想社会などの**新産業**などの創出
- **PCクラスタ(PC cluster)**: 既設のLANあるいはインターネットで安価なパソコンやワークステーションを接続した分散処理環境で実現する**並列処理システム**
- **グリッドコンピューティング(grid computing)**: インターネット接続した複数のコンピュータによって計算サービス, ライブラリ提供, **データベース管理などの機能を分担・協調処理**



**OS機能に影響**

## 【まとめ】 OS機能の発展

### ■ OSの基本機能

#### ● コンピュータシステム(ハードウェア & ソフトウェア)の管理

➤ コンピュータシステム(ハードウェア & ソフトウェア)機能の進化に対応してOS機能や技術そのものも発展

(例)

- UI(ユーザインタフェース): バッチ型OS(例:メインフレーム時代のメインフレームコンピュータのOS) → 会話型OS(例:マイクロプロセッサ時代以降のワークステーションOSとしてのUNIX, パソコンOSとしてのWindows)
- OSが管理するメインメモリ(アドレス)空間: プロセッサの処理幅に依存(8→16→32→64ビット)
- 入出力装置や通信装置: マルチメディア化による多様化, ネットワーク(管理)機能の充実, モバイル(mobile)&ユビキタス(ubiquitous)機能の装備

### ■ OSの応用機能

#### ● コンピュータシステムのユーザ(user; 利用者)に利便性を提供

➤ ユーザインタフェース(user interface; UI)の発展

## (上級)

# ユーザインタフェース(UI)の発展

1. バッチ型OS = バッチ処理OS → 考慮せず

↓ (会話型OSへの移行)

2. 会話型OS = TSS処理用OS → “UI”は未誕生

➤ キーボードと文字ディスプレイ/プリンタによる1次元文字列(テキスト)ベースの入出力インタフェース

↓ (マルチメディア化)

3. グラフィカルUI (graphical UI; GUI)

➤ ポインティング装置(代表例: マウス)とグラフィックディスプレイ(= ビットマップディスプレイ(bitmap display))による図形や静止画像

↓

マルチメディアUI (multimedia UI)

➤ + 音声/動画/映像

## OSの役割(0) (諸定義)

### [定義1.1] ユーザプログラム

- 一般ユーザがコンピュータシステム上で使用する応用プログラム
  - コンピュータシステム上で動くプログラム
    - (a) システムプログラム(=OSプログラム)
    - (b) ユーザプログラム(=OS以外のプログラム)

### [定義1.2] 仮想化

- ソフトウェアによって、実際の/物理的なハードウェア機構/機能を模倣(=シミュレーション(simulation)), 仮想的/論理的機構/機能に見せかけること

# OSの役割 (1)

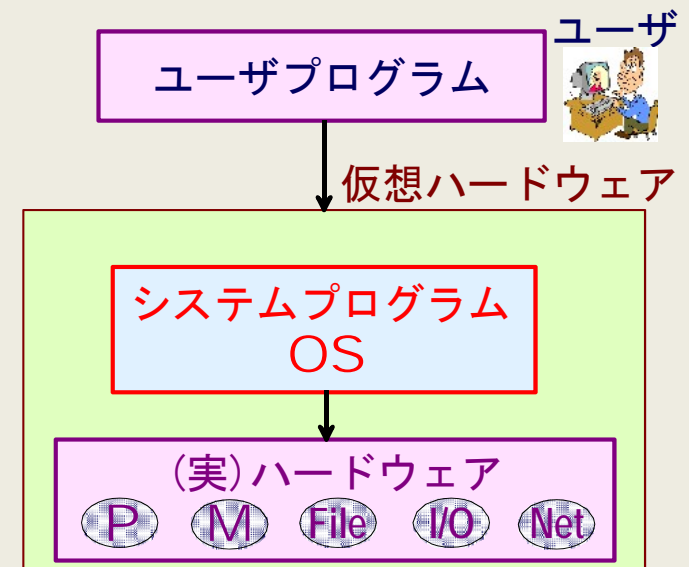
(A) ユーザプログラムから実際の/物理的ハードウェア機構を隠ぺい

↑ (実現)

- OSがハードウェア機構(= プロセッサ/メモリ/入出力装置)を仮想化
  - ハードウェア環境の相違を吸収
  - 統一した・共通な仮想のハードウェア環境をユーザプログラムに提供

[効果] OSさえ共通であれば、ハードウェア環境が異なっても、次を実現

- ユーザプログラムの移植性や互換性を確保 [← ユーザや応用プログラムの開発者の立場]
- ユーザプログラムの開発や保守を支援 [← 応用プログラムの開発者の立場]
- ユーザプログラムを実行 [← ユーザの立場]



## OSの役割 (2)

(B) ユーザプログラムが共用 or 共有ハードウェア (= プロセッサ/メモリ/入出力装置) 資源の **実行時** における管理・制御

↑ (実現)

● ユーザプログラムの実行時にユーザプログラムが使用するハードウェア資源の使用方法を管理・制御

(例) OSが実行時にハードウェア資源の共用/共有が生み出す使用競合を解決, 各種資源を効率的に並行動作

⇔ コンパイラは実行前にハードウェア資源の使用方法を決定・設定

➤ (A)と併せて実現/実現可



## OSの役割 (3)

(C) <sup>こわ</sup>壊れやすいソフトウェア(特にユーザプログラム)を **実行時** (動的)に保護

↑ (実現)

- OSがソフトウェア(= マシン命令やデータ)に対するアクセス(使用)権の管理 & 競合を調停

(例) 不正アクセスの防止, OSとユーザプログラムとの実行時における区別(= 実行モードの設定)