

# ソフトウェアアーキテクチャ

---

環境情報学部

萩野 達也

スライドURL

<https://vu5.sfc.keio.ac.jp/slide/>

# 授業予定

1. オペレーティングシステム
2. ファイルシステム
3. シェルの働き
4. 文書清書システム
5. C言語コンパイラ
6. LISP処理系
7. Java仮想機械
8. ネットワークシステム
9. 名前解決
10. 遠隔利用と電子メール
11. 分散ファイルシステム
12. World Wide Web
13. Windowシステム
14. データベース管理システム

# ソフトウェアの階層

アプリケーション

ライブラリ

ミドルウェア

オペレーティングシステム

BIOS

デバイス ドライバ

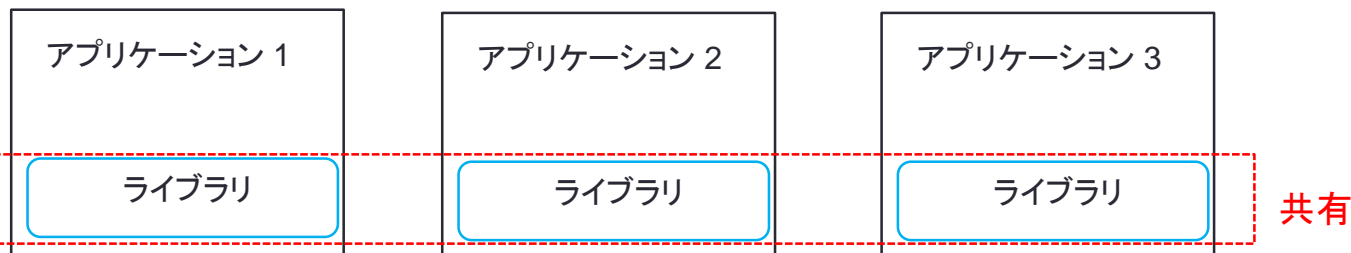
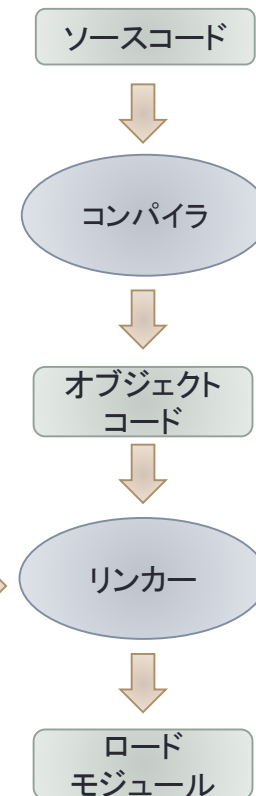
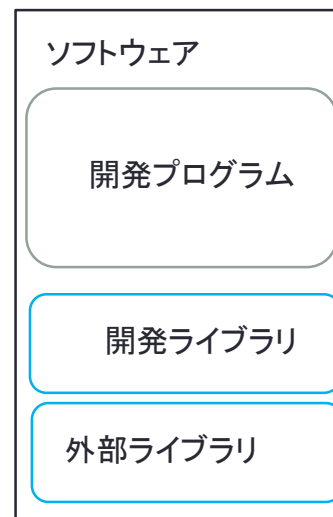
ハードウェア

# アプリケーション

- アプリケーション = 応用
  - ユーザプログラム
  - ユーザが作るプログラムはほとんどアプリケーションプログラム.
- アプリケーションの分類:
  - システムアプリケーション
    - バックアップ
    - メンテナンス
  - 言語処理系
    - コンパイラ
    - インタープリタ
  - 文書, 画像, 動画処理
    - エディタ
  - ネットワークアプリケーション
    - メールソフト
    - Webブラウザ
  - 科学技術アプリケーション
    - 数値解析
    - シミュレーション

# ライブラリ

- ライブラリ
  - 関数や手続きの集まり
  - 単体では動作しない
  - 別のソフトウェアに組み込まれる
- ライブラリの例
  - ファイル操作ライブラリ
  - 数学ライブラリ
  - 統計ライブラリ
  - 文字列処理ライブラリ
- スタティック vs ダイナミック ライブラリ
  - スタティック(静的): コンパイル時に結合
  - ダイナミック(動的): 実行時に連結
- 共有ライブラリ
  - 異なるプログラム間でライブラリを共有する



ソフトウェアアーキテクチャ

# 第1回 オペレーティングシステム

---

環境情報学部

萩野 達也

# オペレーティングシステム

- OSと呼ばれることが多い
  - 基本ソフトウェア
    - ほぼすべてのPCで必要
  - コンピュータで動作するすべてのプログラムを制御する
  - コンピュータの資源を管理する
  - 動作するプログラムに様々な機能を提供する
- 良く知られるOS
  - Windows: 95, 98, Me, NT, XP, Vista, 7, 8, 8.1, 10, 11
  - Mac OS: 9, X, 11, 12, 13, 14
  - UNIX: Linux, FreeBSD, NetBSD, OpenBSD, Solaris
- その他のOS
  - 携帯電話: iOS, Android, Windows phone
  - 実時間システム: iTron, VxWorks

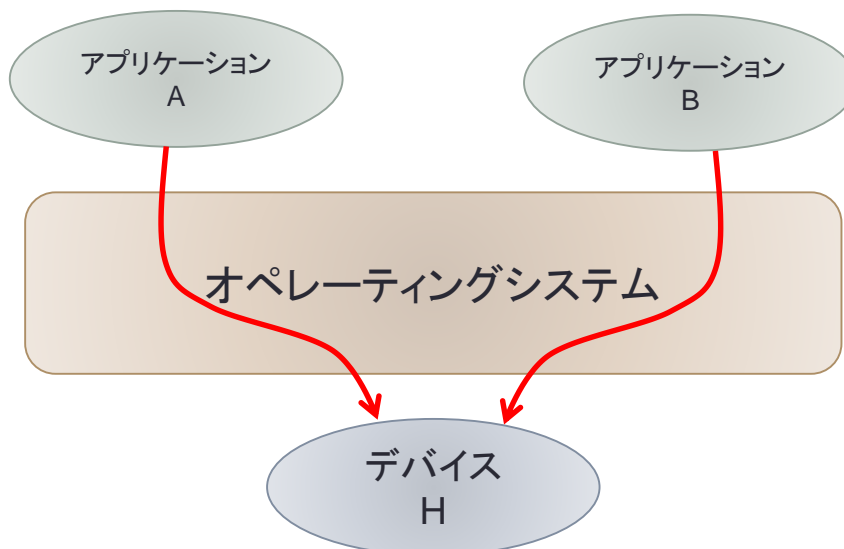
# オペレーティングシステムの役割

- デバイス制御の調停
- プログラム実行環境の分離
- マルチプログラミング
- メモリ管理
- ファイルシステム
- ネットワークシステム
- プログラム間の通信



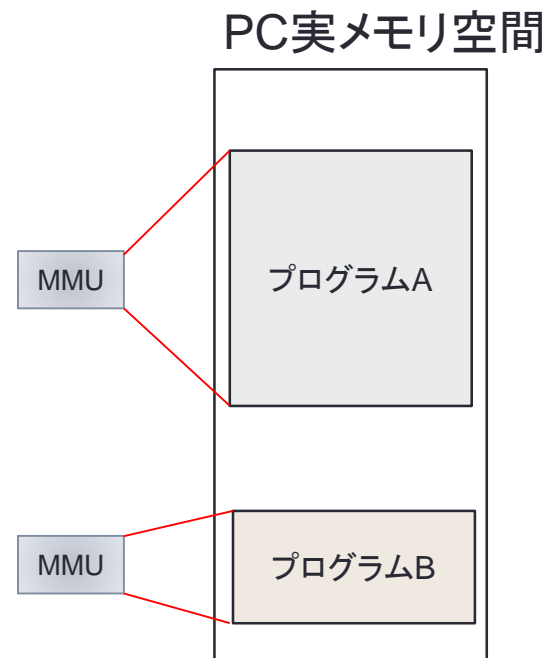
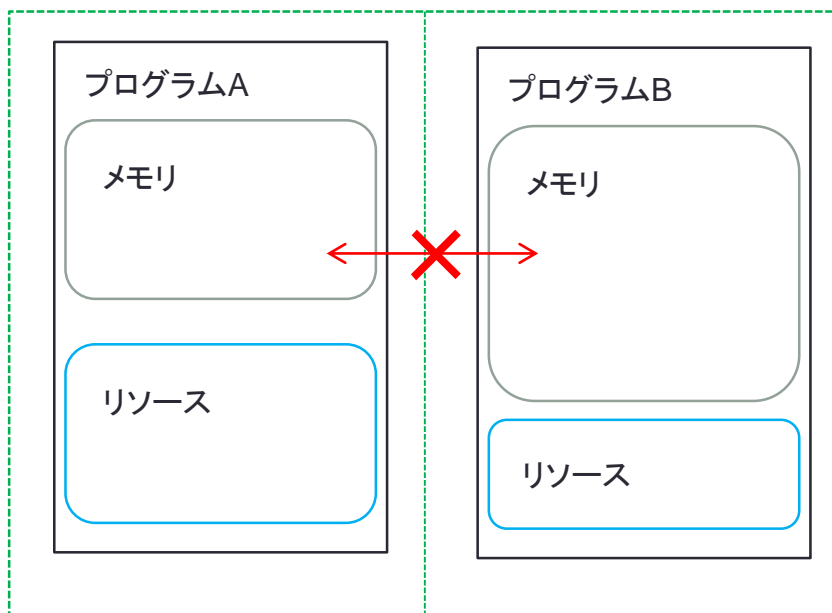
# デバイス制御の調停

- 2つのアプリケーションが同時に同じデバイスを利用してはいけない
  - デバイス=コンピュータに接続しているハードウェア
  - キーボードは1つ, マウスも1つ, プリンタも1台, など
- OSによる調停
  - アプリケーションは直接デバイスを利用しない
  - OSを介してデバイスを利用する



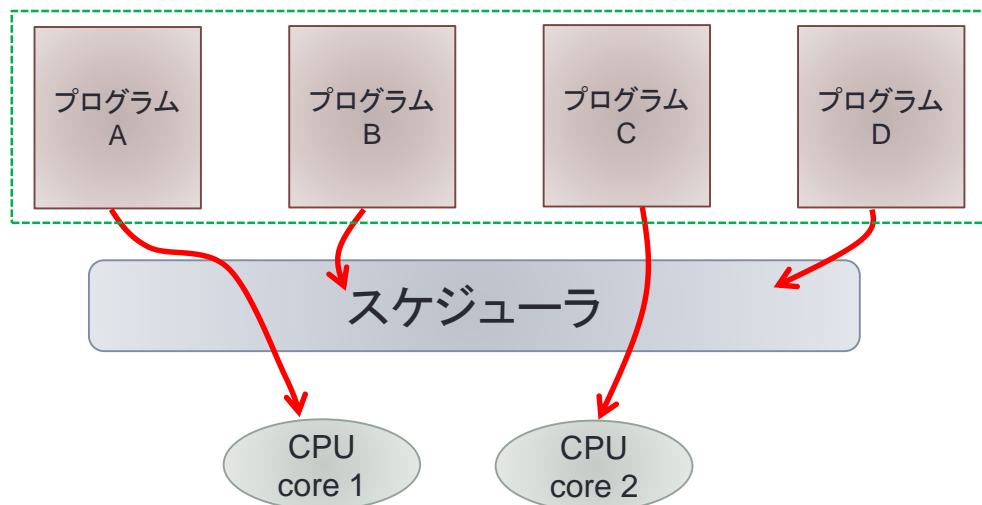
# プログラム実行環境の分離

- プログラムは独立して動作している。
  - 他のプログラムの動作には影響されない。
- それぞれのプログラムは、別々のメモリ空間を使う。
- それぞれのプログラムのメモリ空間は保護されている。
  - 他のプログラムから参照や変更ができない。

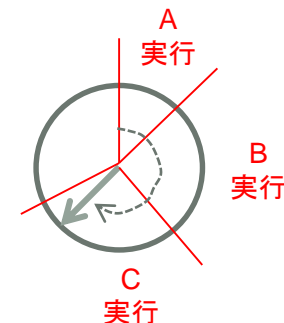


# マルチプログラミング

- 複数のプログラムを同時に実行することができる。
- 同時に実行することができるプログラムの数は、CPUのcore数に制限されない。
- それぞれのプログラムにCPUが割り当てられる。
  - スケジューリング
  - 優先制御

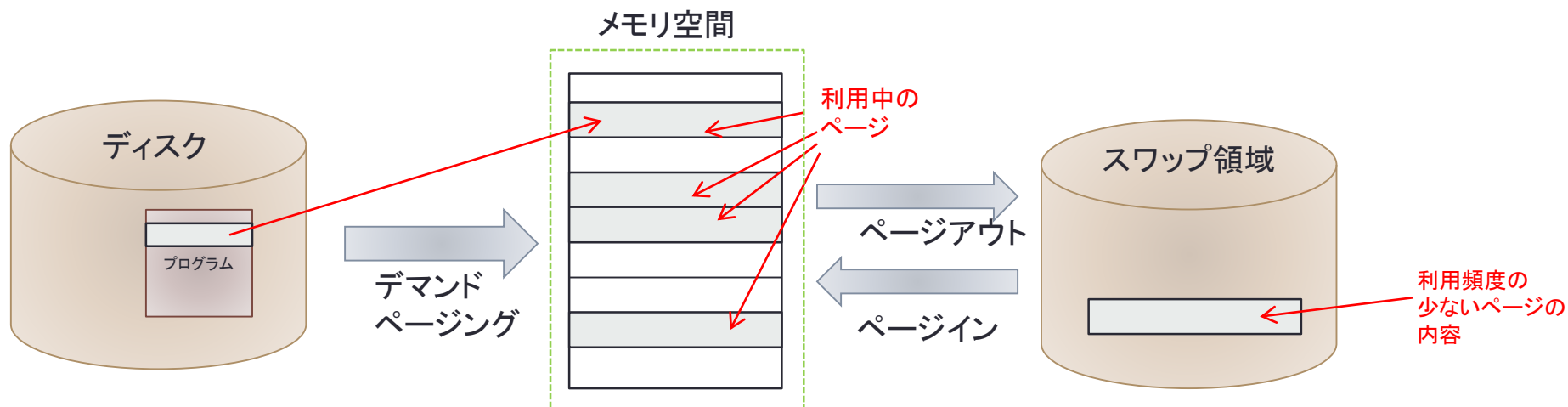


## 時分割



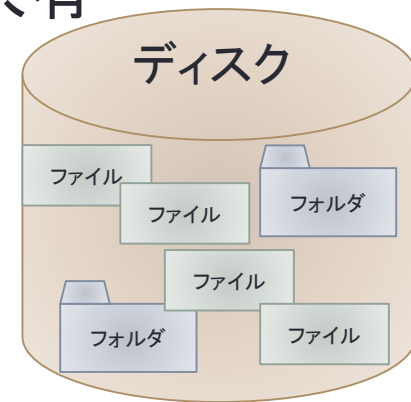
# メモリ管理

- プログラムが必要とするメモリを管理
  - メモリはページ単位(例:1ページ=4KB)に分けられている.
- 不要なメモリを回収し, 必要とするプログラムに割り当てる.
- 仮想記憶の提供
  - ページアウト=利用頻度の少ないメモリ(ページ)は外部記憶(スワップ領域)に追い出し, 空きメモリを増やす.
  - ページイン=必要となった時に, データをメモリ内に読み込みなおす.
  - 実際のメモリの量をプログラムは気にする必要はない.
    - 仮想記憶ではメモリ管理はOSにお任せ
    - 小型ゲーム機などでは, プログラムオーバーレイを使い, 自分でメモリを管理する.
- デマンドページング
  - プログラムやデータは必要となって初めてメモリ内に読み込まれる.
  - プログラム実行開始時はメモリ内は空の状態.



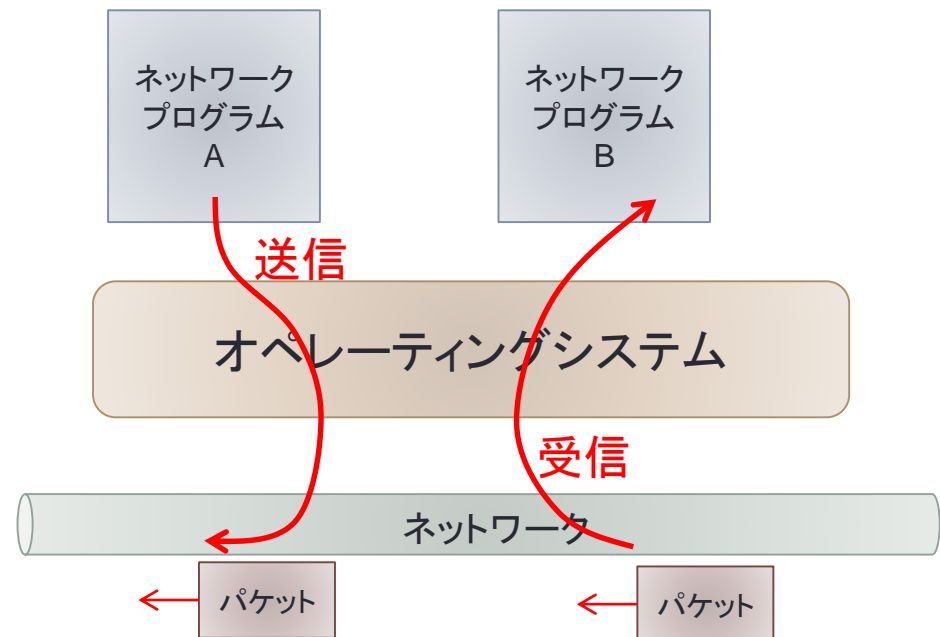
# ファイルシステム

- ディスクなどを複数のアプリケーションで共有
- ディスク上にファイルシステムを構成
- 効率が良いようにバッファリング
- ファイルシステムの種類
  - FAT (Fast Allocation Table)
  - NTFS (NT File System)
  - HFS (Hierarchical File System)
  - UFS (UNIX File System)
  - LFS (Log File System)



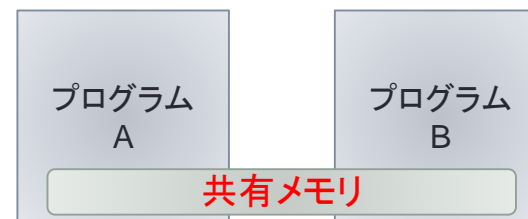
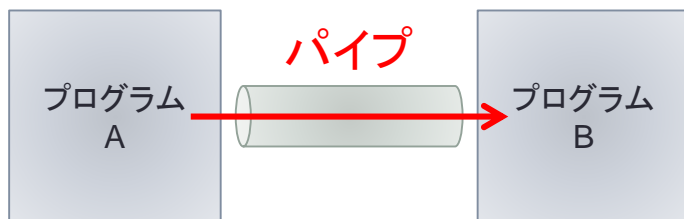
# ネットワークシステム

- ネットワークパケットの処理
  - 送信(プログラムからネットワーク)
  - 受信(ネットワークからプログラムへ)
- TCP/IP 制御
  - フラグメンテーション
  - 再送
  - 順序に並び替え
  - ウィンドウ制御

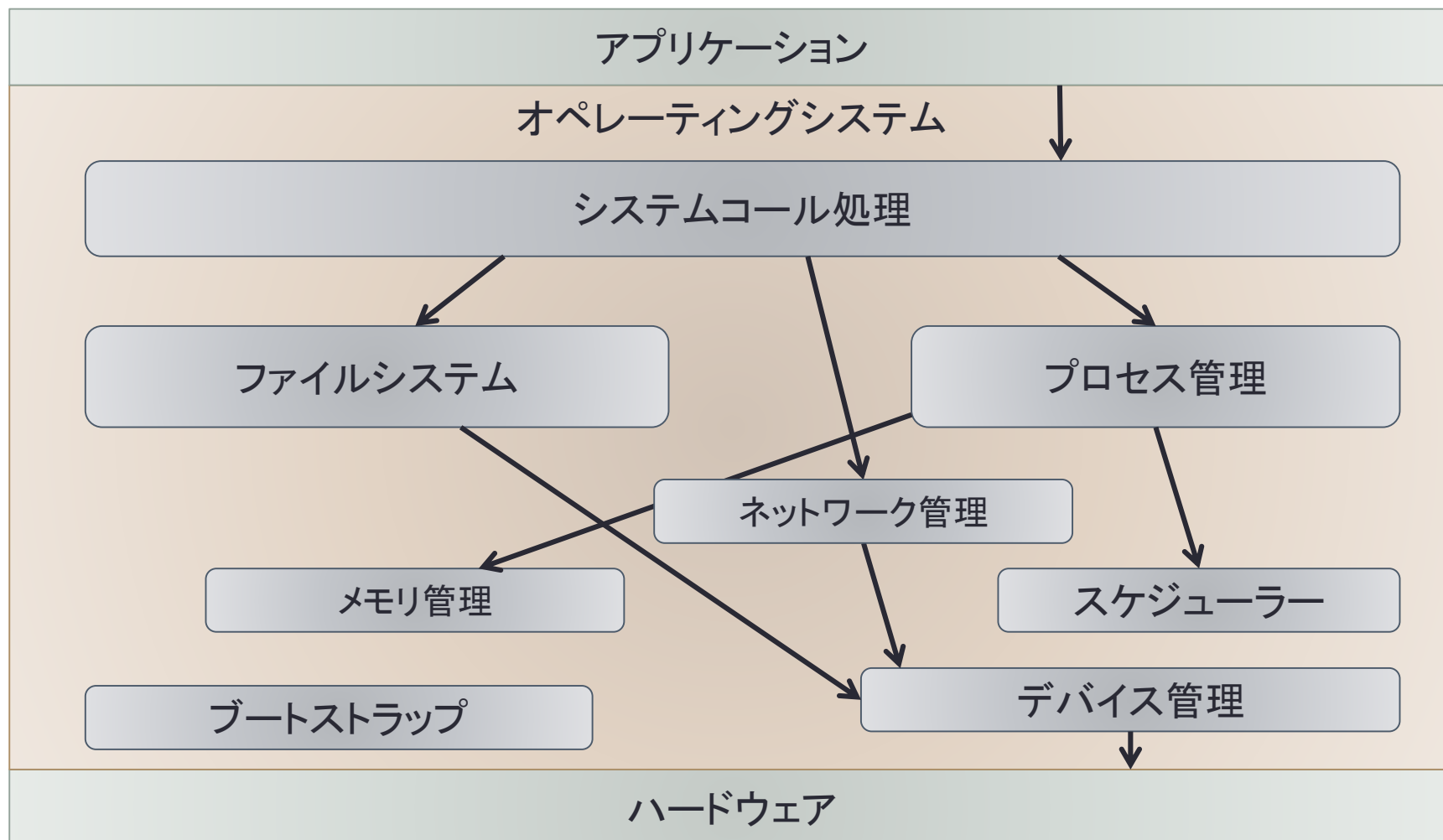


# プログラム間の通信の提供

- プログラム間のデータの受け渡しの仲介
  - パイプ
  - 共有メモリ
  - セマフォ
  - ロック



# オペレーティングシステムの構成要素





# まとめ

- オペレーティングシステム
  - デバイス制御の調停
  - プログラム実行環境の分離
  - マルチプログラミング
  - メモリ管理
  - ファイルシステムの提供
  - ネットワークシステムの提供
  - プログラム間の通信の提供