

情報科学

(7) 基本ソフトウェア

目次

- 基本ソフトウェア(OS)の目的
- 基本ソフトウェアの機能
 - タスク管理
 - メモリ管理
 - ファイルシステム

ソフトウェアの種類と階層

ユーザ

応用ソフトウェア (アプリケーション)
ミドルウェア
基本ソフトウェア (OS)

} システムソフトウェア

ハードウェア

基本ソフトウェア

- OS (Operating System)
- コンピュータのハードウェアの違いを吸収
 - ▣ ユーザ、ソフトウェア、開発者に統一の環境を提供
- 分類
 - ▣ 制御プログラム(狭義のOS)
 - ▣ サービスプログラム
 - ▣ 言語プロセッサ

OSの目的と機能

目的

- 開発効率の向上
- 処理能力の向上
- 信頼性の向上
- 操作性の向上

機能

- 言語プロセッサ
- API
- タスク管理
- メモリ管理
- ファイルシステム
- ネットワーク
- GUI

開発効率の向上

- 言語プロセッサ（コンパイラ etc.）
 - プログラムの記述を助けるソフトウェア
- API（Application Programming Interface）
 - OSの機能を応用ソフトウェアから利用するためのインタフェース
 - = 前のスライドに列挙されているような機能を使うプログラムを作るときに使う

処理能力の向上

- 単位時間当たりの仕事量(スループット)の向上
- Turnaround time 入力開始～出力終了までの時間
- Response time 入力終了～出力開始までの時間



信頼性の向上

- Reliability（信頼性）
 - ▣ 故障や障害が発生しにくいこと（平均故障間隔）
- Availability（可用性）
 - ▣ 高い稼働率を維持できること
- Serviceability（保守性）
 - ▣ 障害復旧のしやすいこと（平均故障時間）
- Integrity（保全性）
- Security（安全性）

操作性の向上

- GUI(Graphical User Interface)
 - ▣ ウィンドウ, アイコン, マウスなど
- マルチタスク
 - ▣ 複数の仕事を並行して実行
 - ▣ 表計算とワープロなど

タスク管理

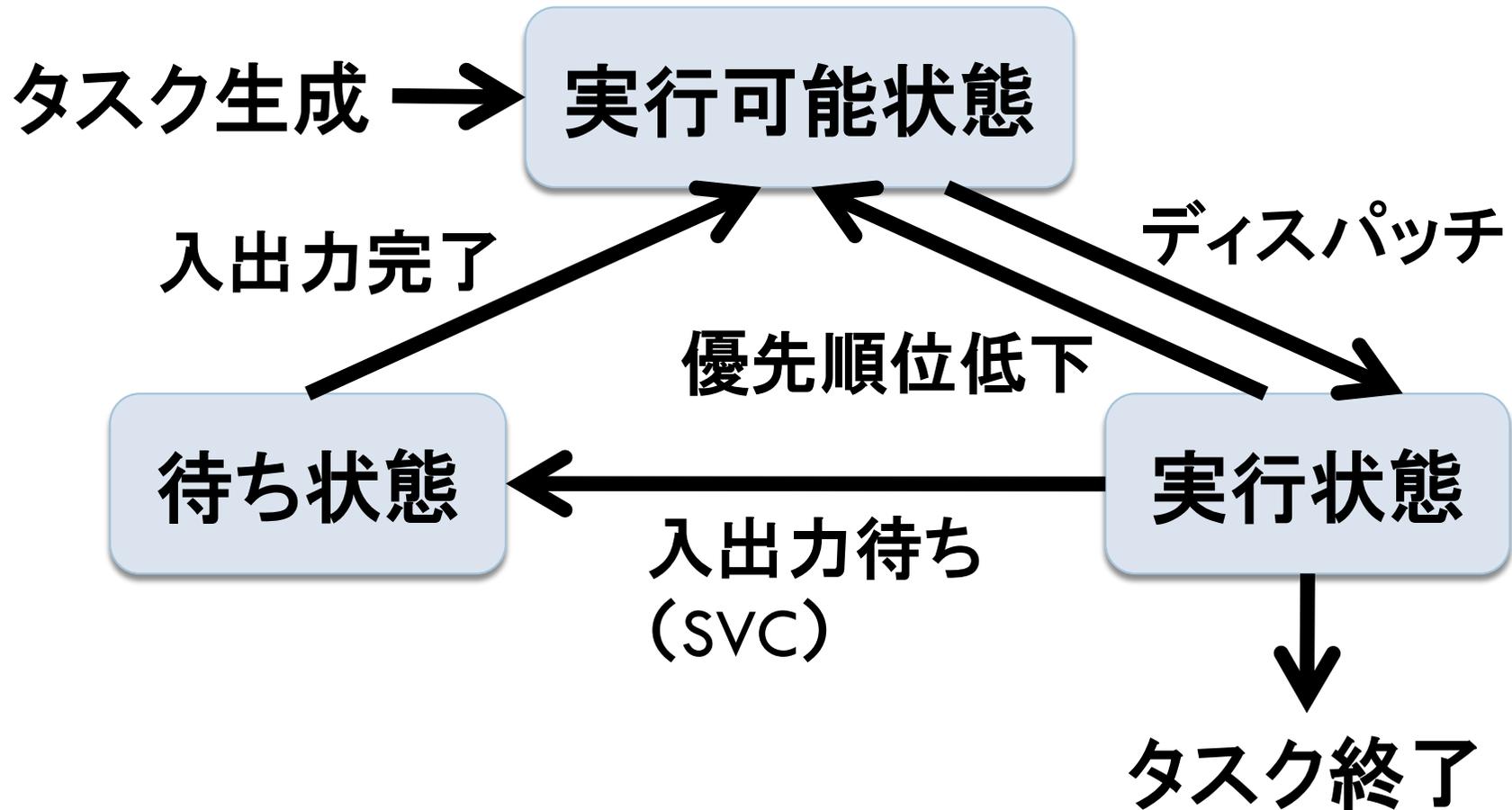
複数のプログラムを交代しながら使えるようにする機能

ジョブとタスク

- ジョブ: 利用者から見た仕事の単位
- タスク: コンピュータから見た仕事の単位
 - ▣ プロセスと呼ぶこともある



タスク管理



ディスパッチ (dispatch)

- 実行可能状態のタスクにCPU使用权を与える
- どのタスクから実行？
 - ラウンドロビン方式
 - 優先度方式

割り込み処理

- 実行中のタスクを中断して他のタスクを実行
- 内部割り込み
 - ▣ プログラム割り込み: ソフトウェアのエラーなど
 - ▣ スーパーバイザコール: CPUを特権モードに切替
- 外部割り込み
 - ▣ 機械チェック割り込み: ハードウェアのエラーなど
 - ▣ 入出力割り込み
 - ▣ タイマー割り込み: 指定した時間経過後に発生

マルチタスク

- 擬似マルチタスク
 - タスクが自分でOSに処理を返す方式
 - 例:タスクが入力待ち状態になった
- プリエンプティブ・マルチタスク
 - 一定時間ごとに異なるタスクを実行
 - タイマーによる強制割り込みを利用

スプーリング

- CPUと入出力装置の速度差を吸収する技術
 - 入出力するデータを一時的に記憶装置に格納
(参考: キャッシュ)
- 印刷や通信で用いられることが多い

メモリ管理

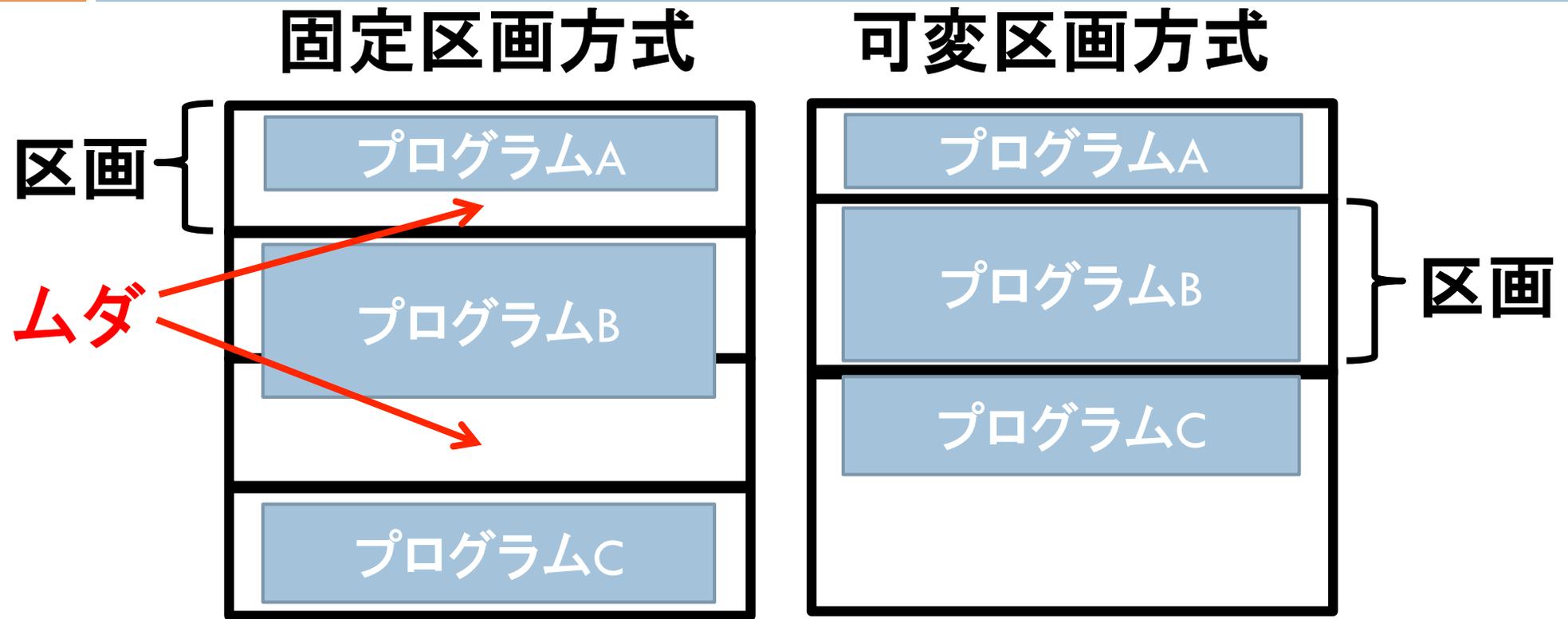
限りあるメモリを重複しないように複数のプログラムに分配してくれる機能

メモリ管理

- プログラムは補助記憶装置にある
- プログラムを実行するには、主記憶に読み込む必要がある
- プログラムの実行状態も主記憶に保持される

- 問題
 - ▣ 複数のプログラムを同時に実行するときは？
 - ▣ 主記憶よりも大きいプログラムを実行するには？

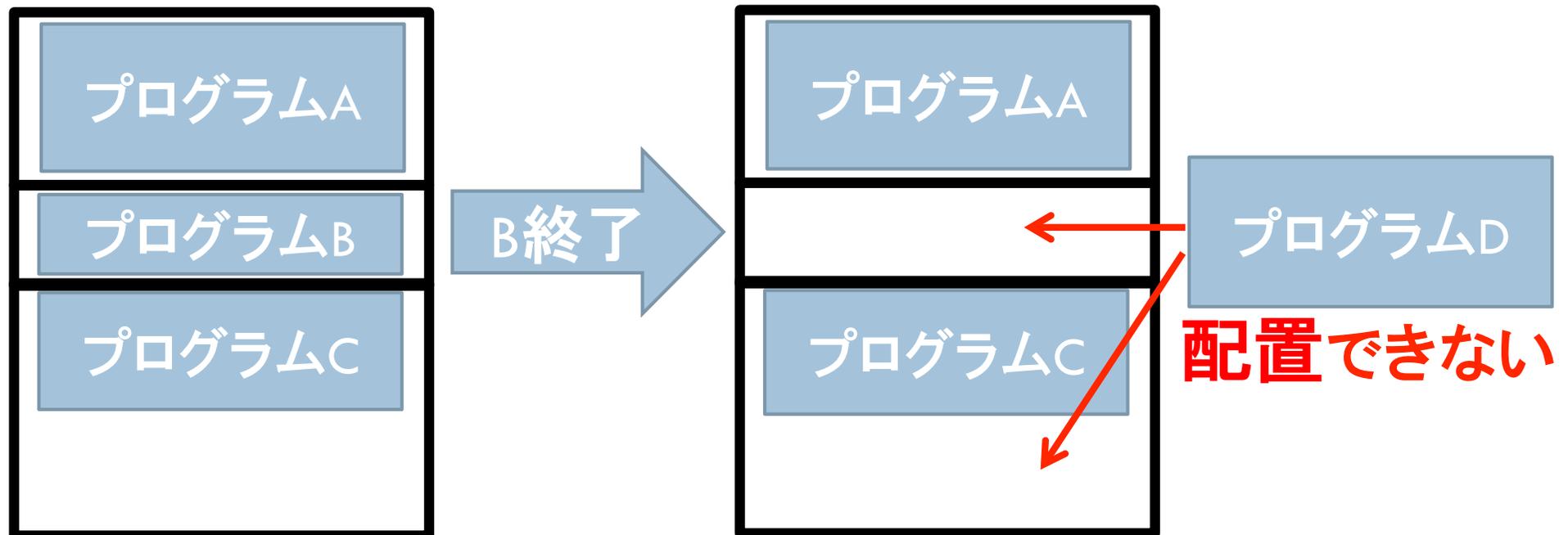
固定区画方式と可変区画方式



- 固定区画方式では、区画内にムダが発生
= 内部断片化

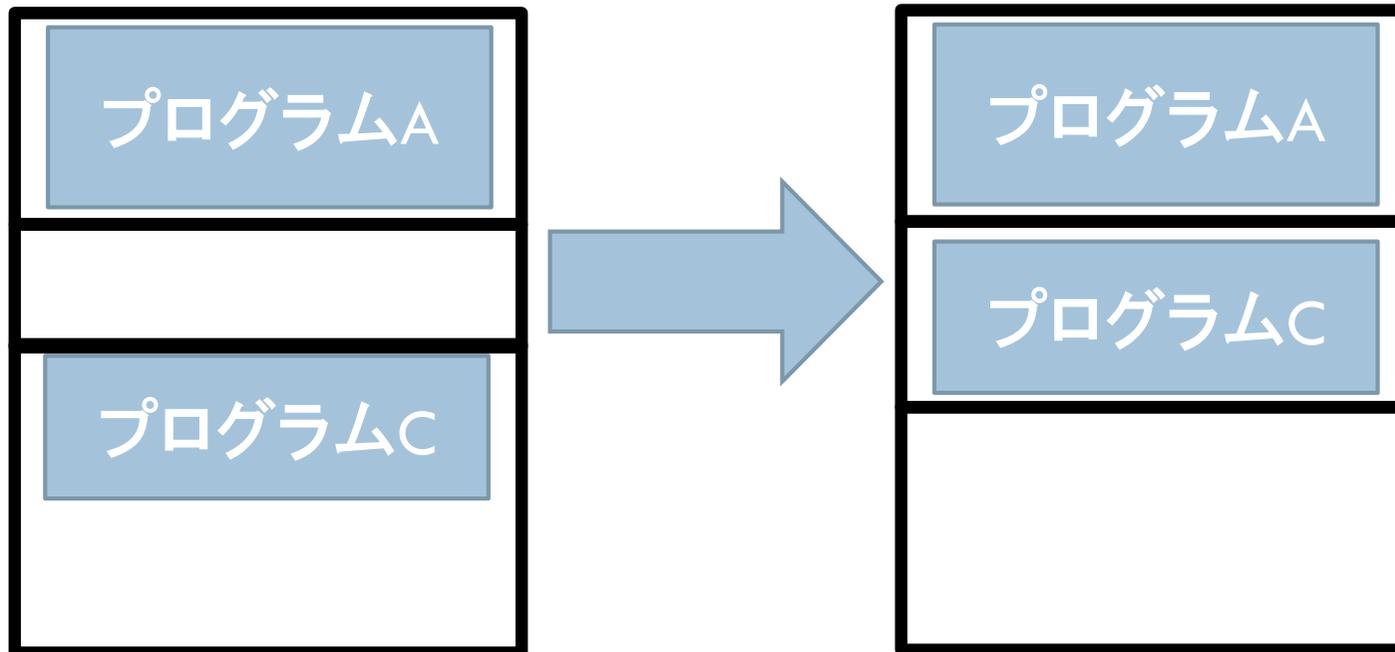
外部断片化(可変区画方式)

- プログラムの実行と終了を繰り返すと発生
=外部断片化



メモリコンパクション

- プログラムの配置を変更して断片をまとめる



それでも足りない！

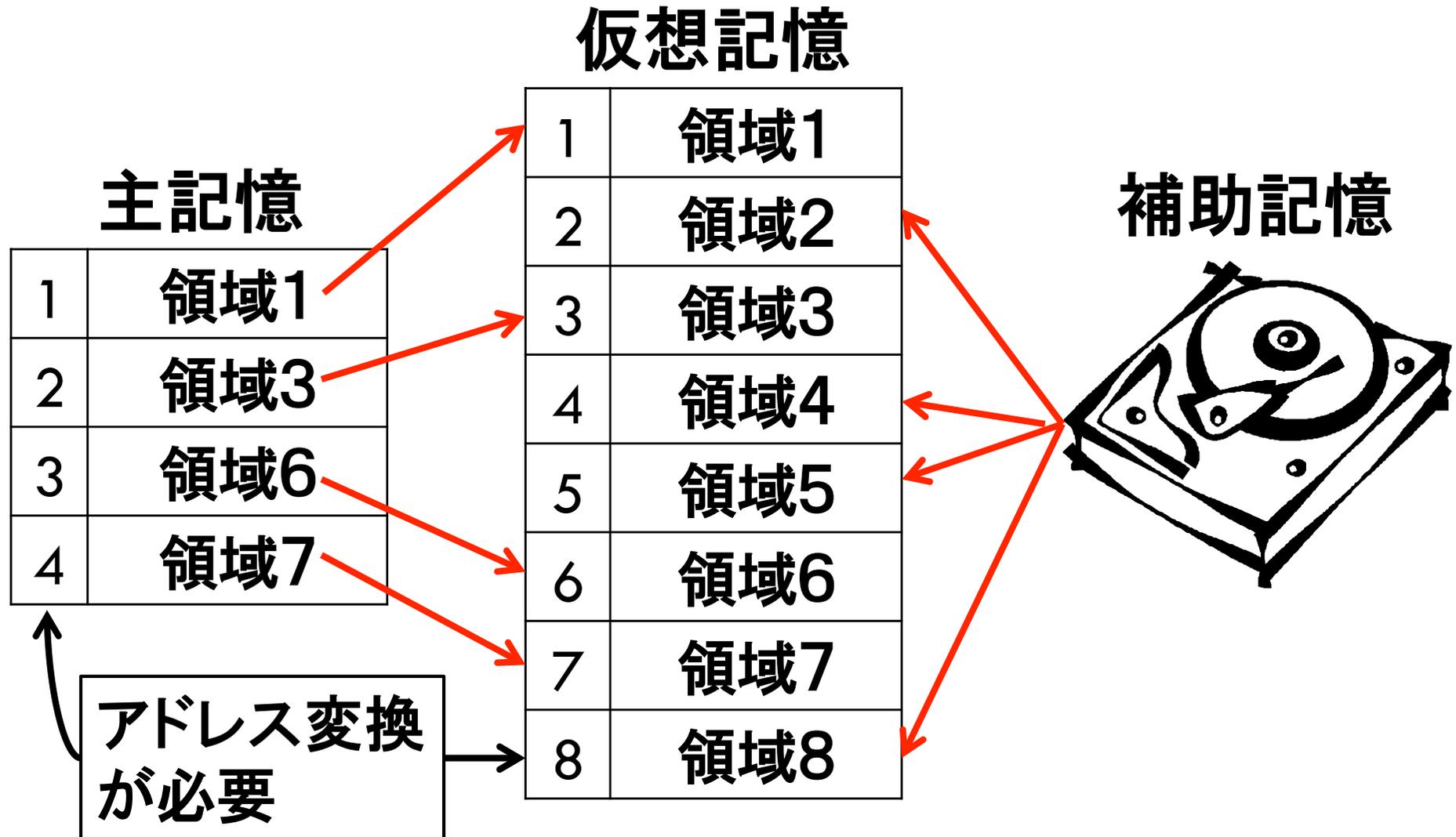
- スワップ
 - ▣ 待ち状態のプログラムを補助記憶に退避
- オーバレイ
 - ▣ プログラムの今実行している部分だけ読みこむ
- 仮想記憶

仮想記憶

ハードウェア	基本ソフトウェア	アプリケーション
主記憶	物理メモリ	仮想記憶領域
補助記憶	スワップ領域	
	ファイル領域	ファイル領域

- 主記憶の容量より大きい仮想の記憶領域を提供

仮想アドレスとアドレス変換



仮想記憶の実現

- 動的アドレス変換(DAT)
 - 仮想記憶アドレスから実記憶アドレスへの変換
(ハードウェア)
- 仮想記憶の実現方式
 - ページング方式: 区画サイズ固定
 - セグメント方式: 区画サイズ可変

ページング方式

- 記憶領域をページ単位に分割
- 実行に必要なページを主記憶上に読み込み
(ページイン)
- 主記憶に空きスペースがない場合は置き換え
ページを決定
- 実行に必要なのないページを補助記憶に退避
(ページアウト)

ページング方式における動的アドレス変換

- 仮想ページ番号と物理ページ番号の対応表
- 上位ビット→対応表を引く
- 下位ビット→そのまま



置き換えページの決定

- FIFO (First-In-First-Out)
 - ▣ 主記憶上に最も古く存在するページ
- LIFO (Last-In-First-Out)
 - ▣ 主記憶上で最後に参照されたページ
- LRU (Least Recently Used)
 - ▣ 未参照時間の最も長いページ
- LFU (Least Frequently Used)
 - ▣ 参照頻度の最も少ないページ

スラッシング

- ページの置き換えが頻繁に起きて処理効率が著しく低下する現象
- 原因: 実記憶が不足している

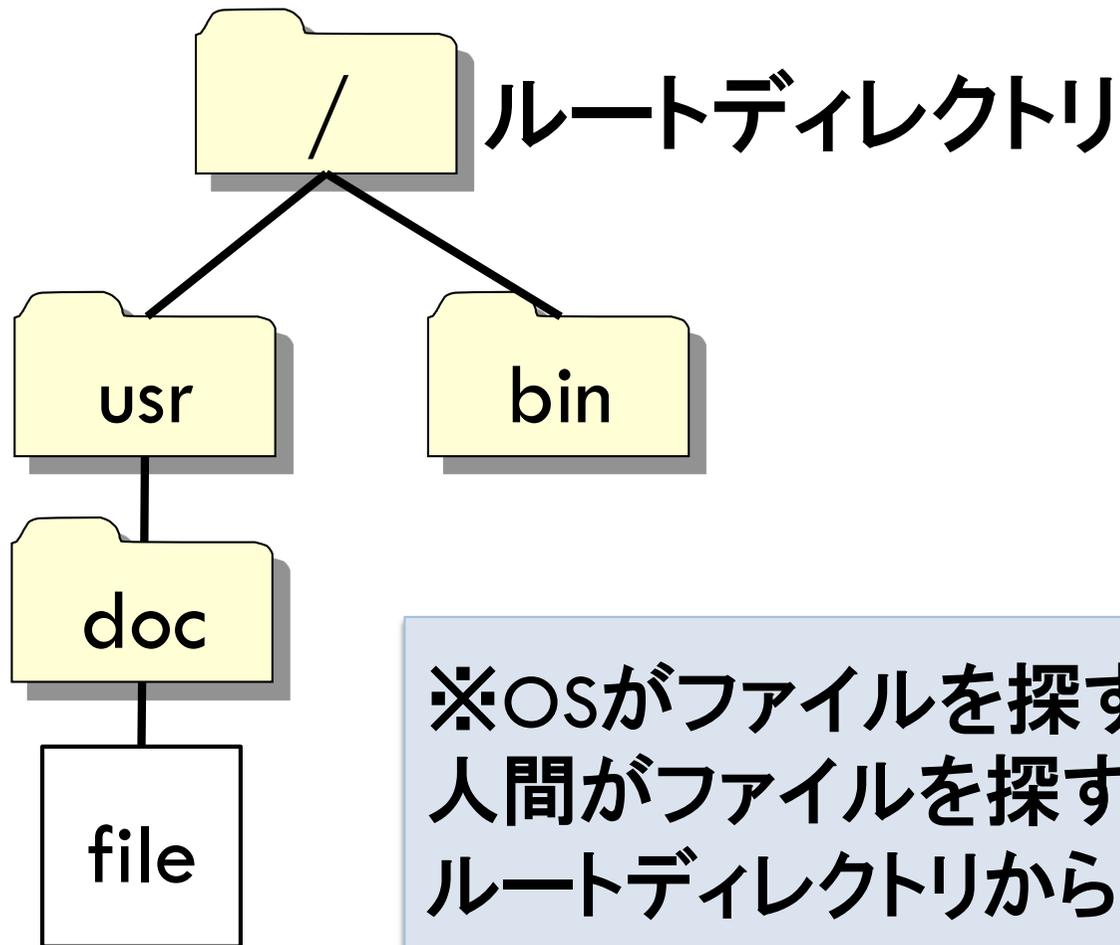
ファイルシステム

ハードディスク上でファイルを整理整頓してくれる機能

ファイルシステム

- ハードディスク上のファイル配置を管理
- 2つの表を使って管理
 - ▣ ディレクトリ（ディレクトリ・テーブル）
 - ▣ FAT (File Allocation Table)

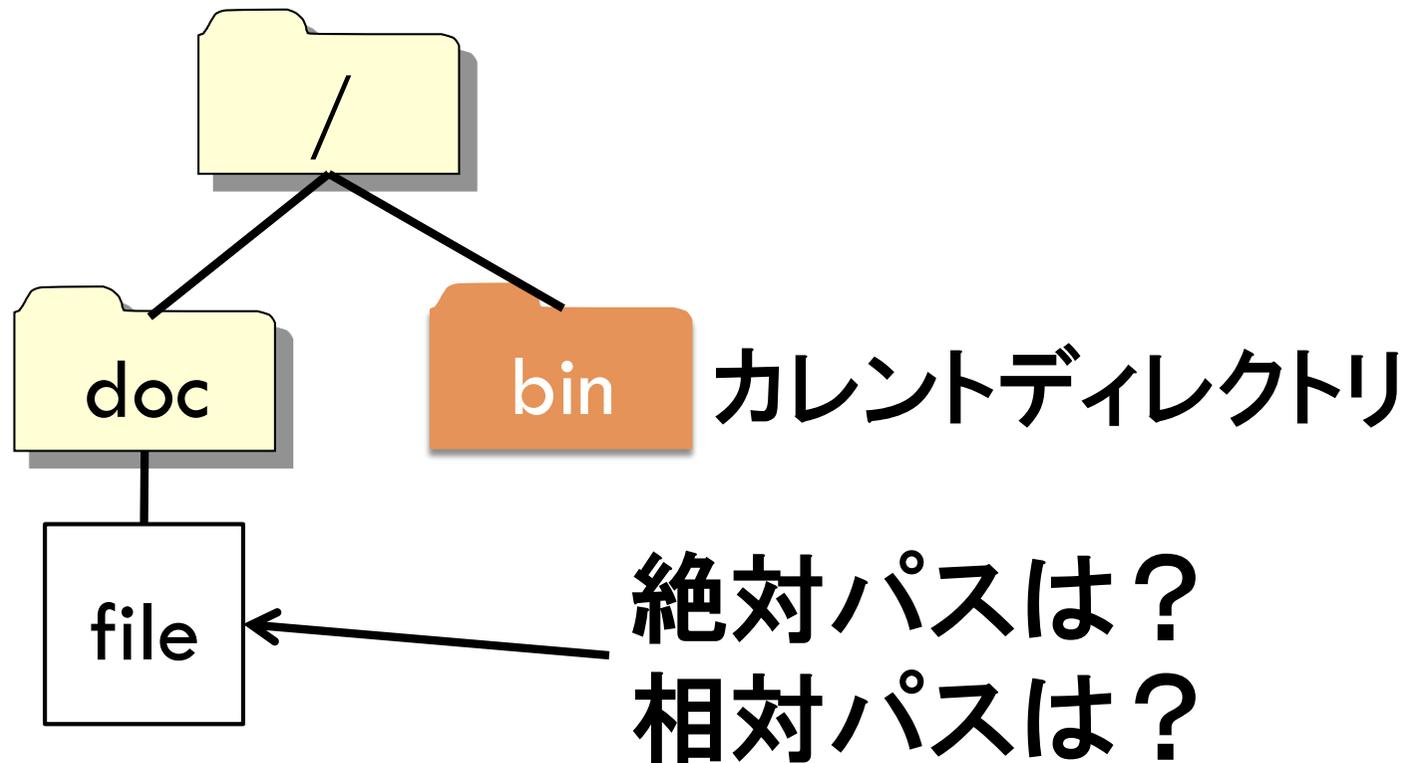
ディレクトリ階層によるファイル管理



※OSがファイルを探すときには、人間がファイルを探すときと同じようにルートディレクトリから順に辿って見つける

ファイル位置の指定

- 絶対パス指定: ルートディレクトリからの位置
- 相対パス指定: カレントディレクトリからの位置



ファイルの操作

- データ領域・ディレクトリ・FATを変更
 - 書き込み
 - コピー
- ディレクトリとFATのみ変更
 - 削除



言語プロセッサ

言語処理プログラム

- アセンブラ
 - ▣ アセンブリ言語の原始プログラムを機械語に変換
- インタプリタ
 - ▣ 高級言語の原始プログラムを1命令ずつ実行
- コンパイラ
 - ▣ 高級言語の原始プログラムを機械語に変換

プログラミング言語

- 低水準言語
 - コンピュータが理解しやすい言語
 - 機械語, アセンブリ言語
- 高水準言語(高級言語)
 - 人間が理解しやすい言語
 - C, FORTRAN, PASCAL, Java, Perl, Ruby など

コンパイラ

