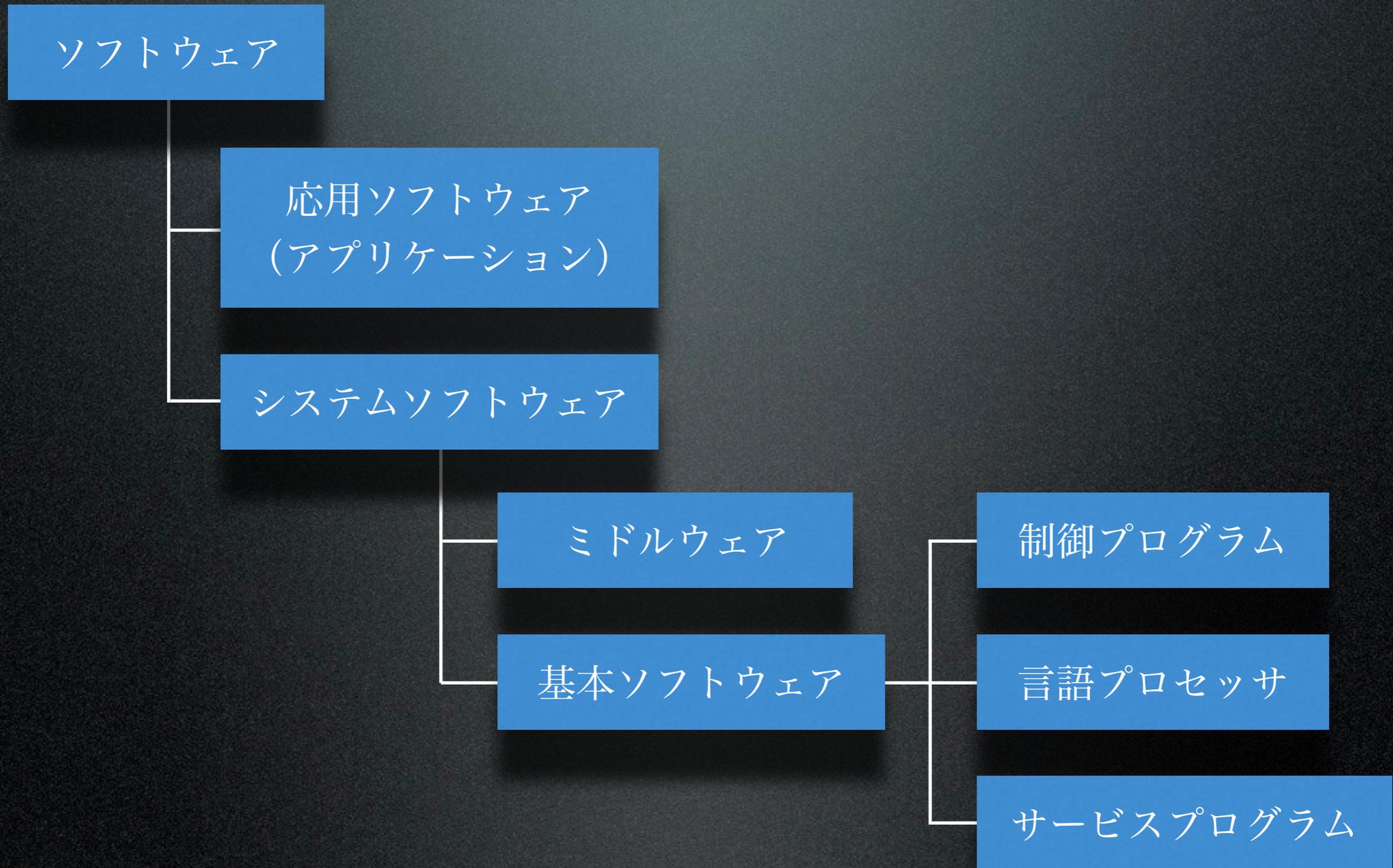


情報科学

(4) 基本ソフトウェア

ソフトウェアの種類



ソフトウェアの階層

ユーザ

応用ソフトウェア

ミドルウェア

基本ソフトウェア

ハードウェア

基本ソフトウェア

- OS (Operating System) とも言う
- コンピュータのハードウェアの違いを吸収
→ ユーザ, 開発者, ソフトウェアに対して統一した環境を提供

OSの目的

- ハードウェア資源の有効活用
- 処理能力の向上
- 信頼性の向上
- 開発効率の向上
- 操作性の向上

処理能力の向上

- Throughput : 単位時間あたりの仕事量
- 応答時間
 - Turnaround Time : 主にバッチ処理
 - CPU時間 + 入出力時間 + 処理待ち時間
 - Response Time : 主にオンライン処理
 - 入力終了から出力を得始めるまでの時間

信頼性の向上

- Reliability (信頼性)
 - システムが正常に稼働していること, MTBF (平均故障間隔)
- Availability (可用性)
 - いつでもシステムが使用可能であること, 稼働率
- Serviceability (保守性)
 - 故障の早期発見, 早期修復ができること, MTTR (平均故障時間)
- Integrity (保全性)
 - 誤作動がなく, システムが壊れにくい
- Security (安全性)

開発効率の向上

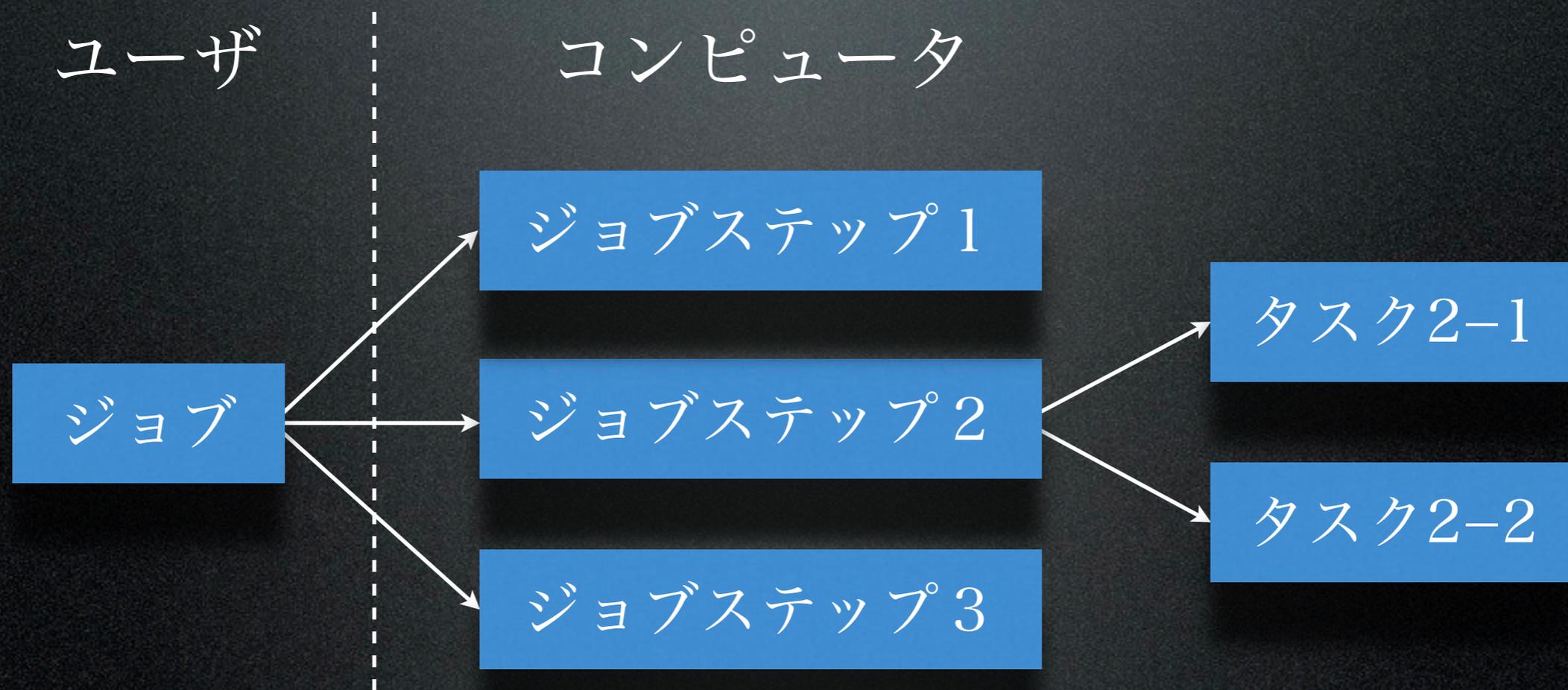
- API (Application Program Interface)
 - OSへの仕事依頼窓口
- Framework

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
 - 複数の仕事を平行して実行
 - 表計算とワープロなど

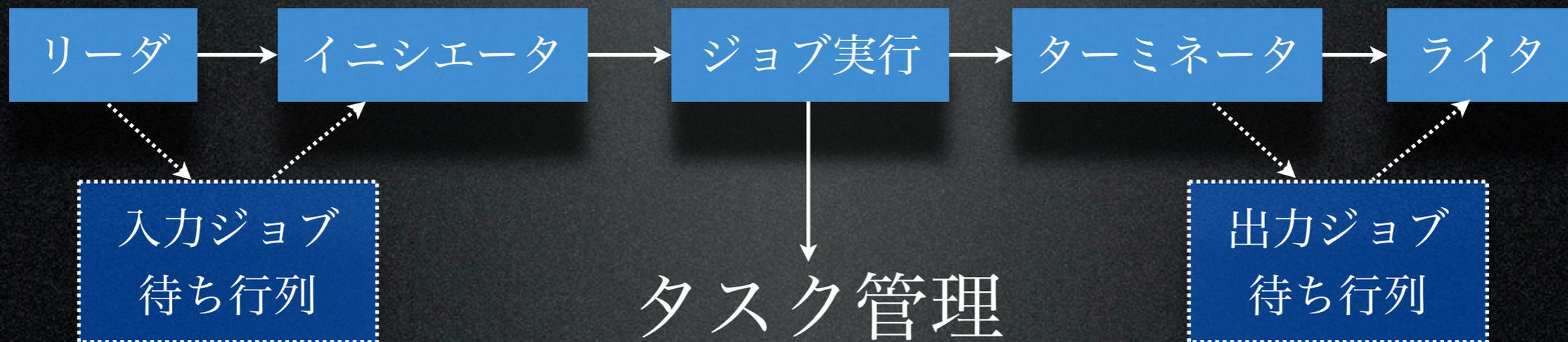
ジョブとタスク

- 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

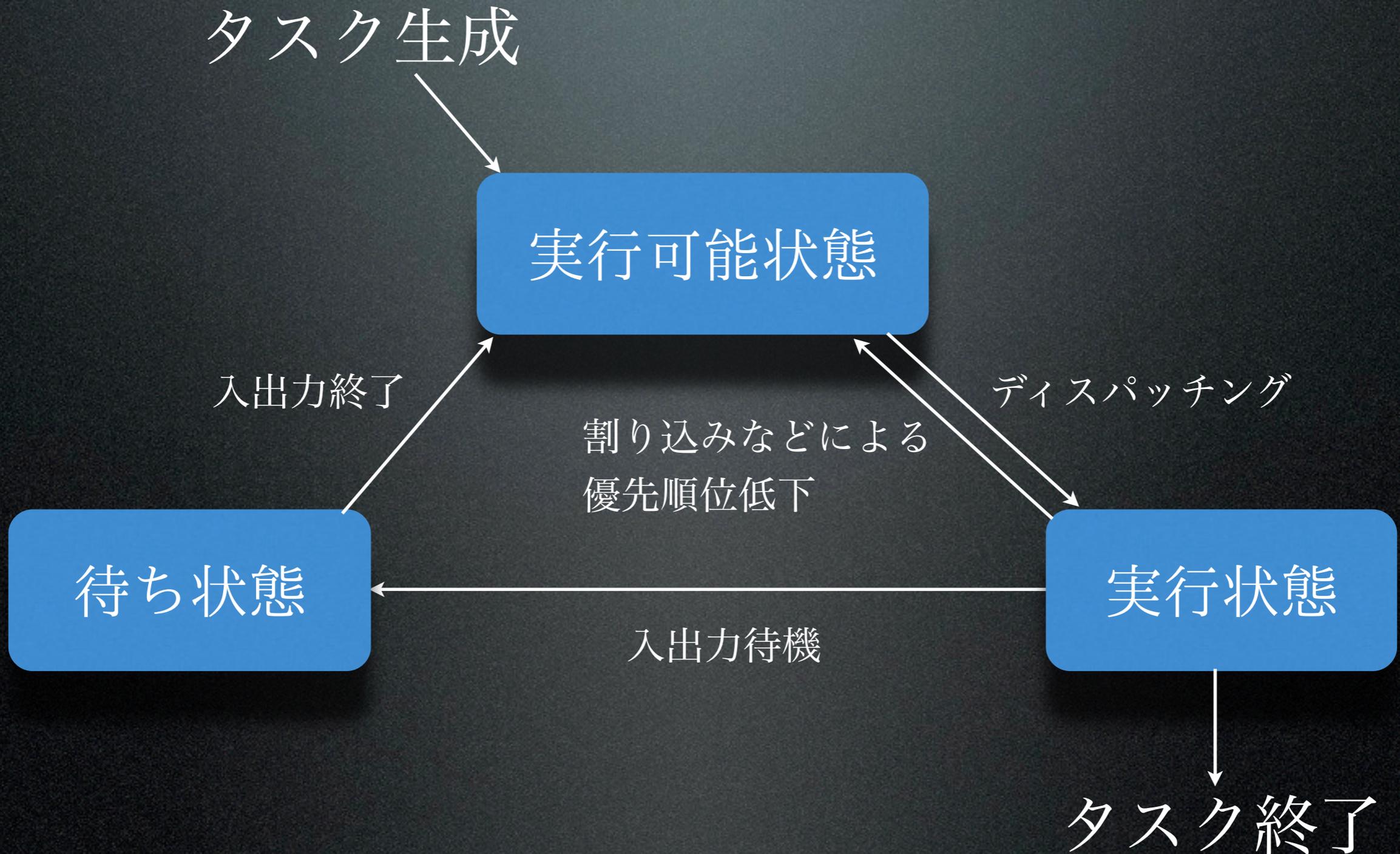


ジョブ管理

- マスタスケジューラ
 - オペレータとジョブスケジューラを仲立ち
- ジョブスケジューラ
 - ジョブステップの実行順を管理



タスクの状態遷移



タスク管理

- ディスパッチング
 - マルチタスク (多重プログラミング)
- 割り込み処理
- スプーリング

ディスパッチング

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

マルチタスク

- 擬似マルチタスク
 - タスクが待ち状態 → 他のタスクを実行
 - 見かけ上の複数タスクの同時実行
- 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

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
- 割り込みの種類
 - 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のモード切替など
 - 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

スプーリング

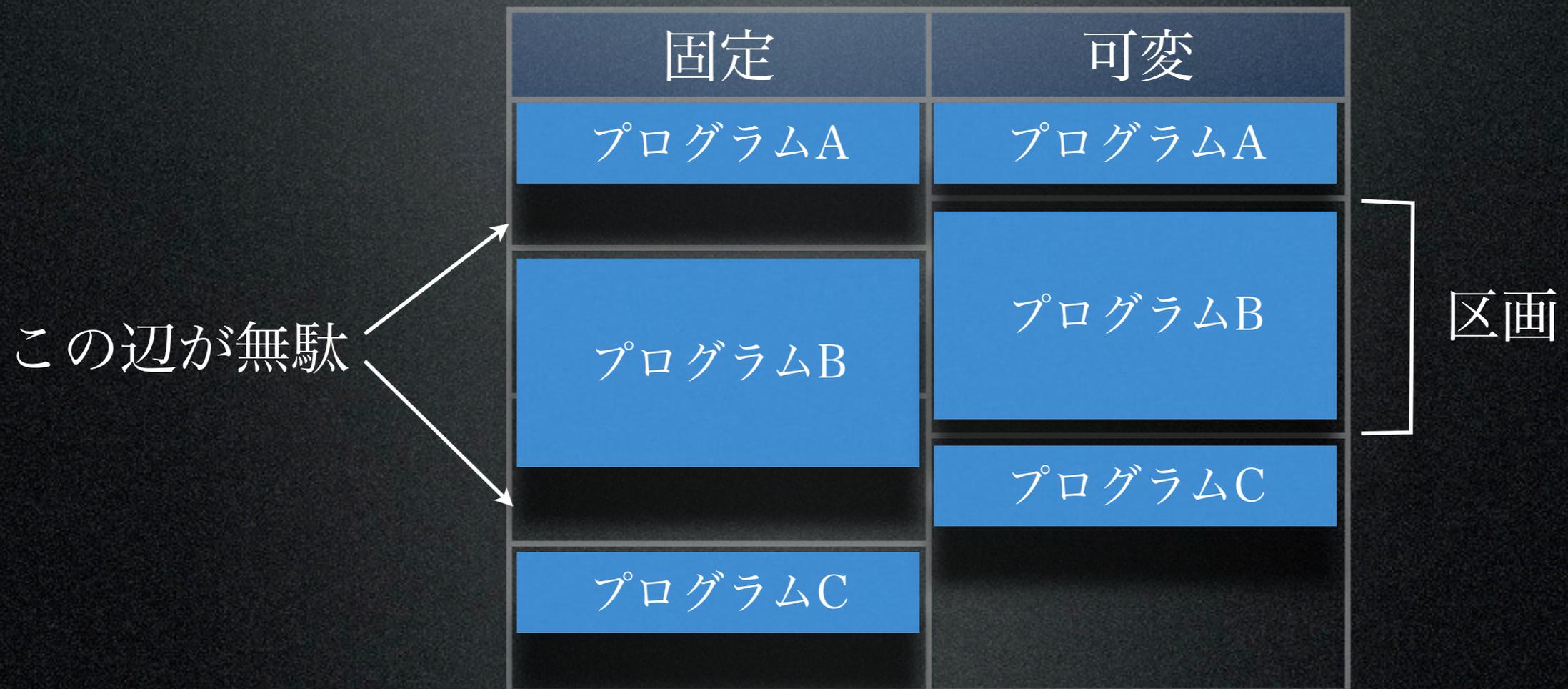
- CPU（高速）と入出力装置（低速）のつなぎ
- 入出力データをバッファリング
 - キャッシュに近い考え方
- プリンタや通信機器などに多い

記憶管理

ハードウェア	基本ソフトウェア	応用ソフトウェア
主記憶	実記憶	ソフトウェアの 記憶領域
補助記憶	仮想記憶	
	ファイル領域	ファイル領域

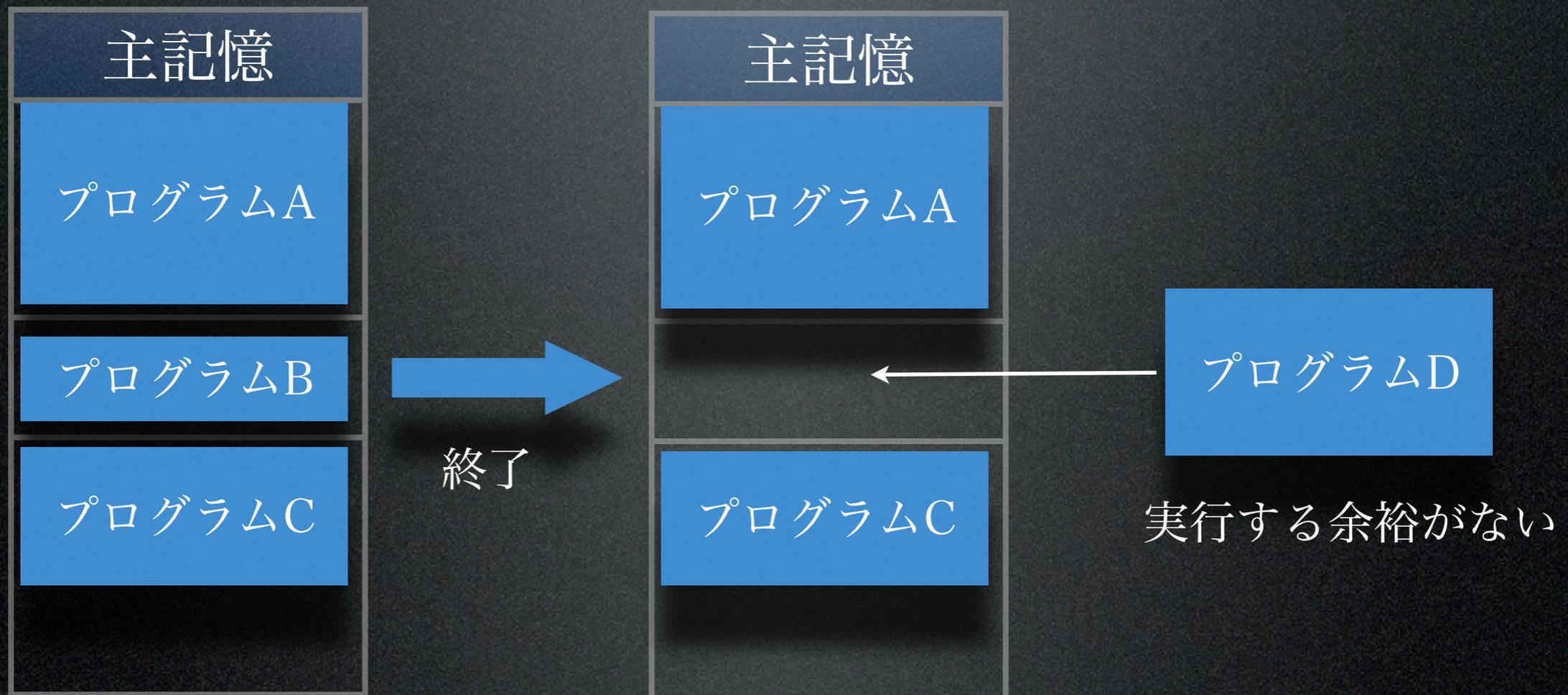
実記憶管理

- 区画方式：主記憶を区画に分けて管理
 - 固定区画方式：区画サイズは固定
 - 可変区画方式：区画サイズは可変



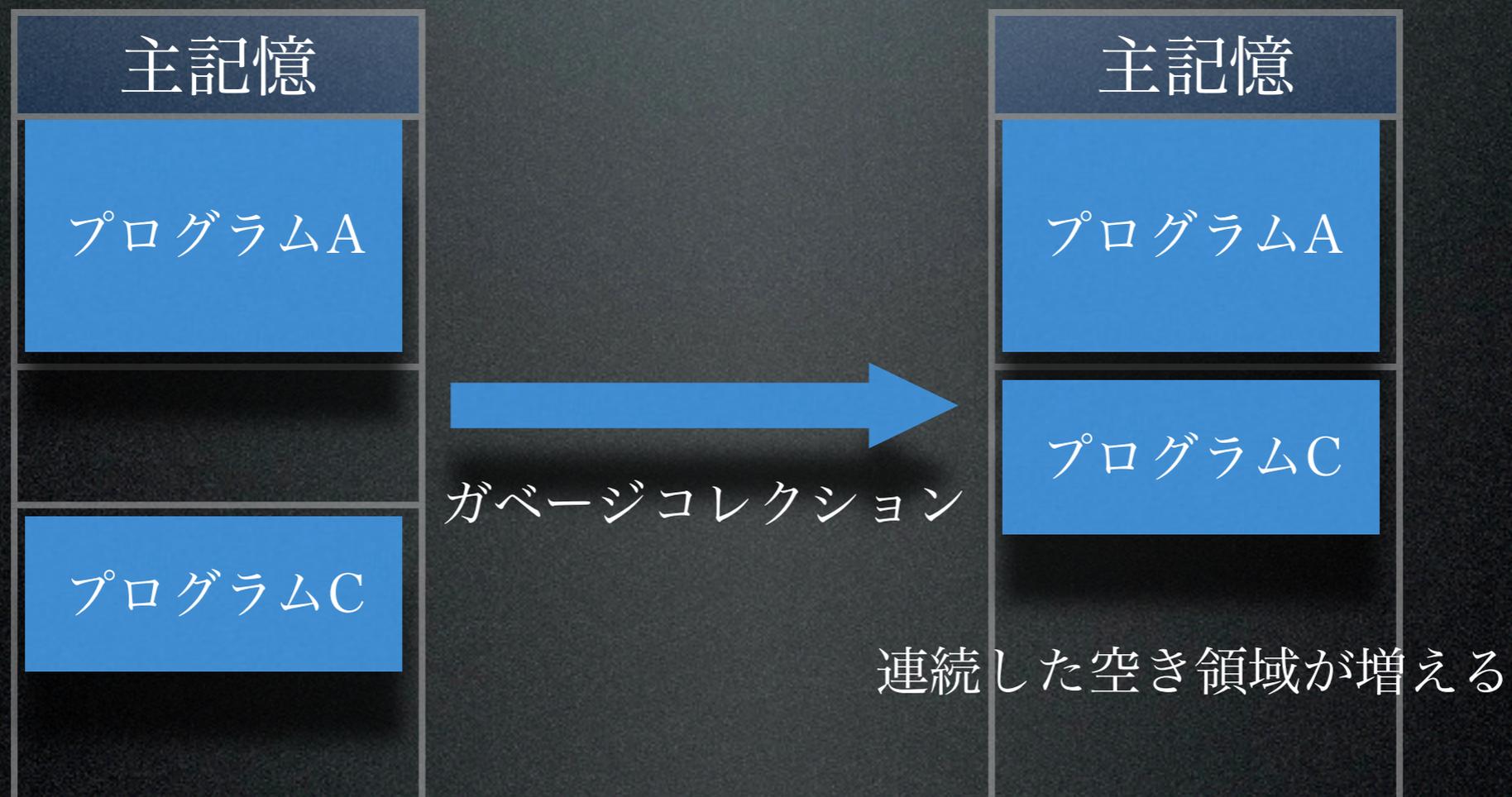
主記憶のフラグメンテーション

- 主記憶の断片化
 - プログラムの実行と終了を繰り返すとできる



フラグメンテーションの解消

- ガベージコレクションまたはメモリコンパクション
 - どちらも同じこと
 - 主記憶内でプログラムの位置を変更し無駄をなくす



プログラムの配置方式

- 再配置可能 (relocatable)
- 再入可能 (reentrant)
- 再使用可能 (reusable)
- 再帰的 (recursive)

オーバーレイ方式

- 主記憶上の共通部分は共用する
 - 共通部分が重複しない → メモリの無駄遣い対策
- 共有ライブラリ方式
 - WindowsのDLLなど

スワッピング

- プログラムを主記憶と補助記憶の間で移動
 - 主記憶 → 補助記憶 (スワップアウト)
 - 補助記憶 → 主記憶 (スワップイン)

仮想記憶管理

- 動的アドレス変換
 - 仮想記憶アドレスと主記憶アドレスの変換
 - 仮想記憶の実現
 - 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
- ＊ 実記憶管理の区画と混乱しないように注意

ページング方式

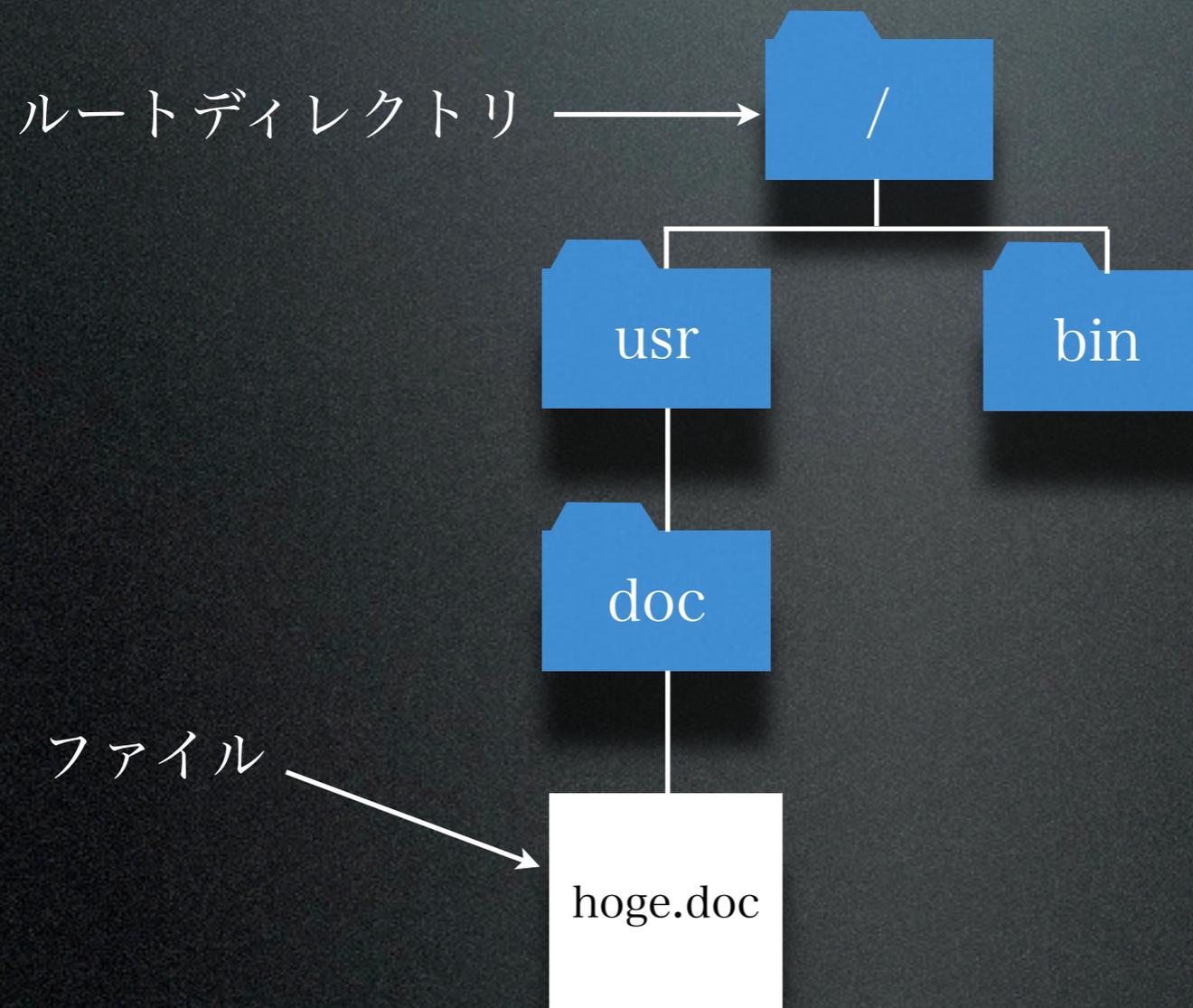
1. 記憶領域をページ単位に分割
2. 実行に必要なページを主記憶上に読み込み (ページイン)
3. 主記憶に空きスペースがない場合は置き換えページを決定
4. 実行に必要なのないページを補助記憶に退避 (ページアウト)
5. 2に戻る

主記憶の置き換えページ決定

- 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

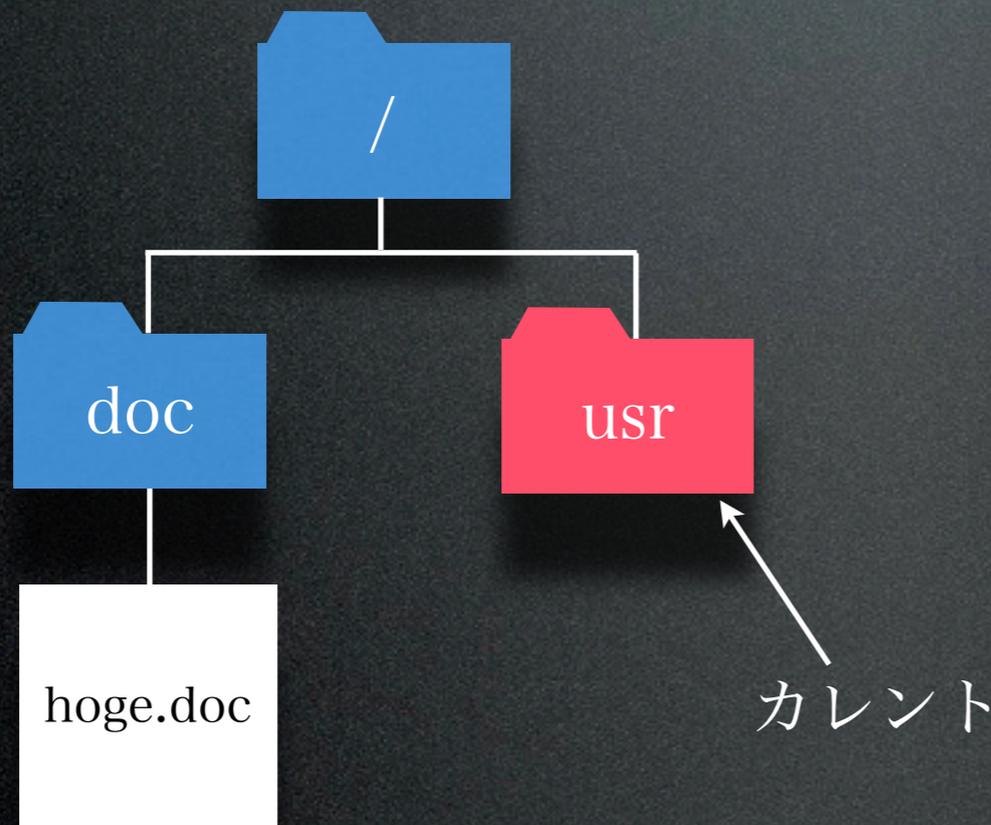
ファイル管理

- ディレクトリ階層によるファイル管理
 - UNIX系OSとそれに倣ったOSが採用



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を記述
- 相対パス指定：カレントディレクトリからの位置を記述
 - カレント～：現在注目しているディレクトリ



– 絶対パス指定
`/doc/hoge.doc`

– 相対パス指定
`../doc/hoge.doc`

カレント

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

言語プロセッサ

- 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

コンパイラ

- 現在最もよく利用されている言語プロセッサ

