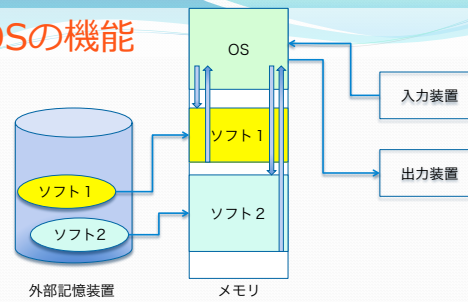


## この章で学習すること

- 5.1 オペレーティングシステムの歴史と目的
  - OSの歴史, 変遷 OSの目的
- 5.2 オペレーティングシステムの基礎
  - OSのソフトウェア構成
  - パソコン用OSの歴史と種類, 機能の違いについて
- 5.3 マルチプログラミングと割り込み
  - 複数のプログラムを同時に実行する仕組みについて
  - 割り込み (内部割り込み, 外部割り込み)
- 5.4 OSの機能
  - ジョブとタスク, スレッド
  - タスクの状態遷移と割り込みの関係
- 5.5 記憶管理 (省略)
- 5.6 言語プロセッサ
  - コンパイルとリンク, ロードモジュール

## OSの機能



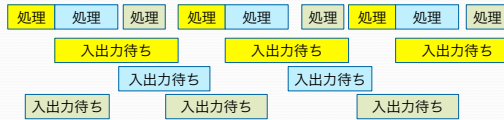
- ソフトウェアを外部記憶装置から読み込む
- 複数のソフトウェアをメモリ上に配置して実行する
- 入出力装置への仲介を行う

## OSの構成

- 制御プログラム
  - コンピュータ上のソフトウェアの動作を管理・調整するプログラム。
    - タスク管理
    - 入出力装置へのアクセスの仲介 (抽象化)
- 言語プロセッサ
  - 高級言語 (C言語など) のコンパイルを行う。
- サービスプログラム
  - OSに付属するソフトウェア。
  - ユーティリティともいう。
  - ファイルのコピー, 閲覧などユーザの計算機利用を便利にするプログラム。

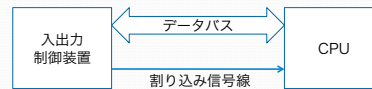
## マルチプログラミング

- 1台の計算機で, 見かけ上, 複数のプログラムを同時に実行すること。
  - CPUは1個しかないと, 実際には, ある瞬間にはどれか1つのプログラムしか動いていない。
  - しかし, プログラムの実行を高速に切り替えれば, ユーザには同時に実行されているように見える。
  - 入出力待ちの間に他の処理をすることができる。

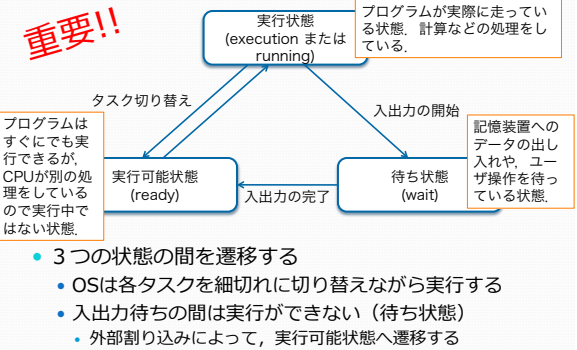


## 割り込み (外部割り込み)

- 計算機が処理中に, 緊急の処理を行う仕組み
  - 入出力制御装置からCPUへ, 事態の発生を知らせる
    - 入出力処理の完了, ユーザの操作
    - 時間の経過 (タイマ)
    - 故障, 不具合などの発生
  - 通常, 割り込みが発生すると, 動作中のプログラムは即座にOSに戻される
    - OSは割り込みの種類を判定し, それに対処する

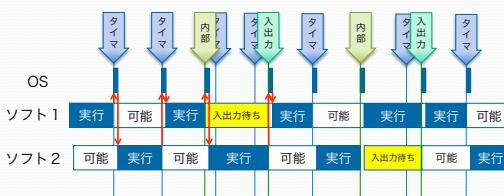


## タスクの状態遷移



## タスク管理

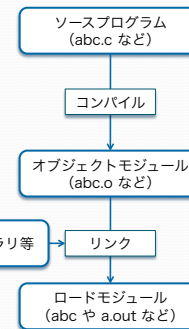
- タイマ割り込み (外部割り込み) によって定期的にOSに制御を戻す
- ソフトは入出力を内部割り込みによって行う
- 入出力が完了すると, 入出力完了割り込み (外部割り込み) によって処理を切り替える



## プログラムの作成手順

重要!!

- 実行可能プログラムを作るまで
  - 高級言語の **ソースプログラム** は, **コンパイラ** によってオブジェクトモジュールに変換される
  - 複数のオブジェクトモジュールとライブラリ等 (printf などの関数が入っている) を **リンク** (linker) がつなぎあわせて, **実行可能プログラム** (ロードモジュール) が出来る
  - 複数のソースプログラムに分けて開発ができる
  - 他人が作成したライブラリを利用することができる



## インタプリタとコンパイラ

- インタプリタ (interpreter)
  - 高級言語の記述を, 解釈しながら実行する
  - 事前のコンパイルが不要
  - 解釈時間が必要なため, 処理が遅い
  - BASIC が代表例。
  - 近年はスクリプト言語 (Perl等) でよく使われる。
- コンパイラ (compiler)
  - 事前に, 高級言語から機械語に変換する
  - 処理速度が速い。
  - 事前に文法チェックなどを行うことが出来る。
  - C言語など. 大規模なソフトやOSはコンパイラで作る。