

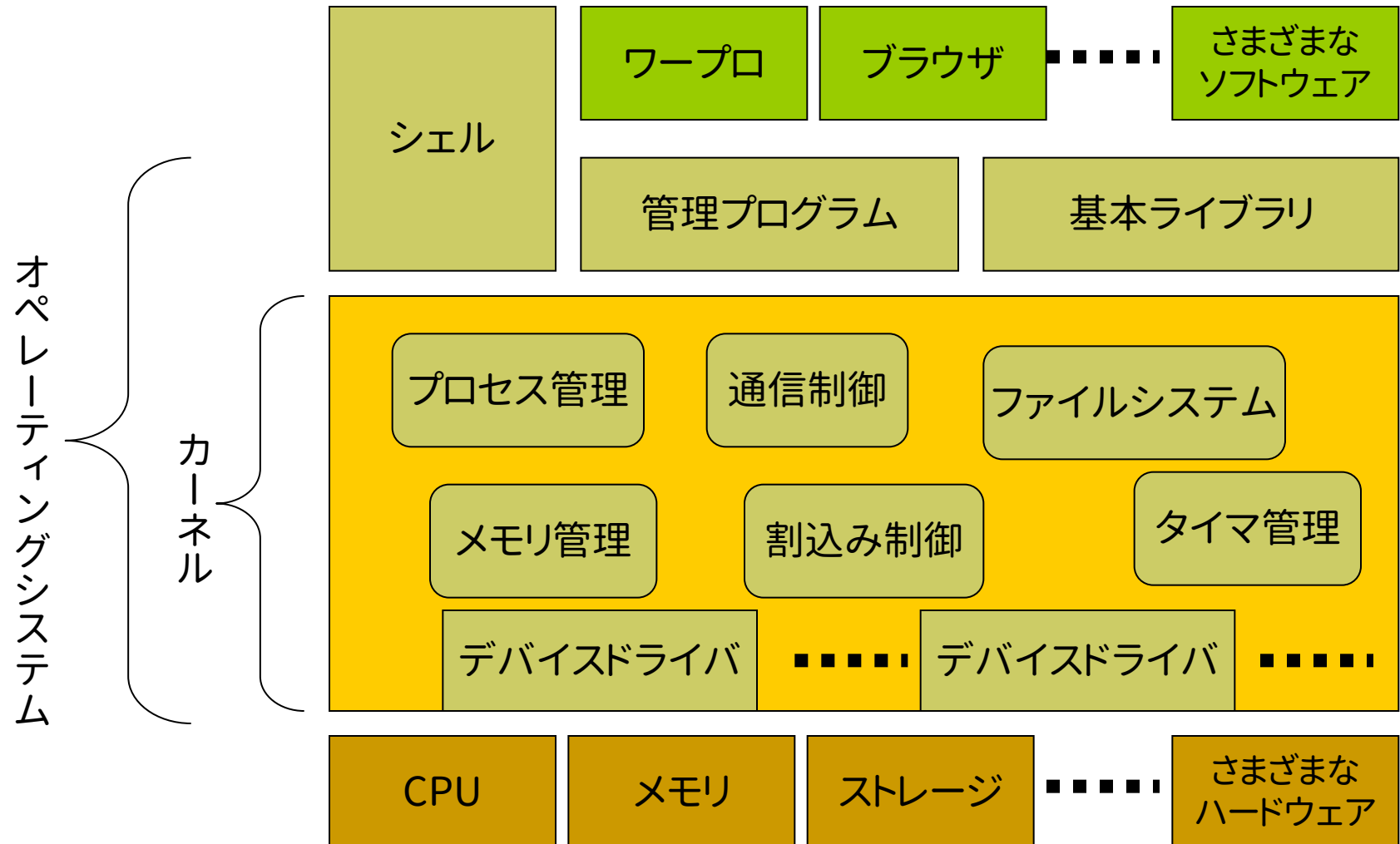
# Operating Systems



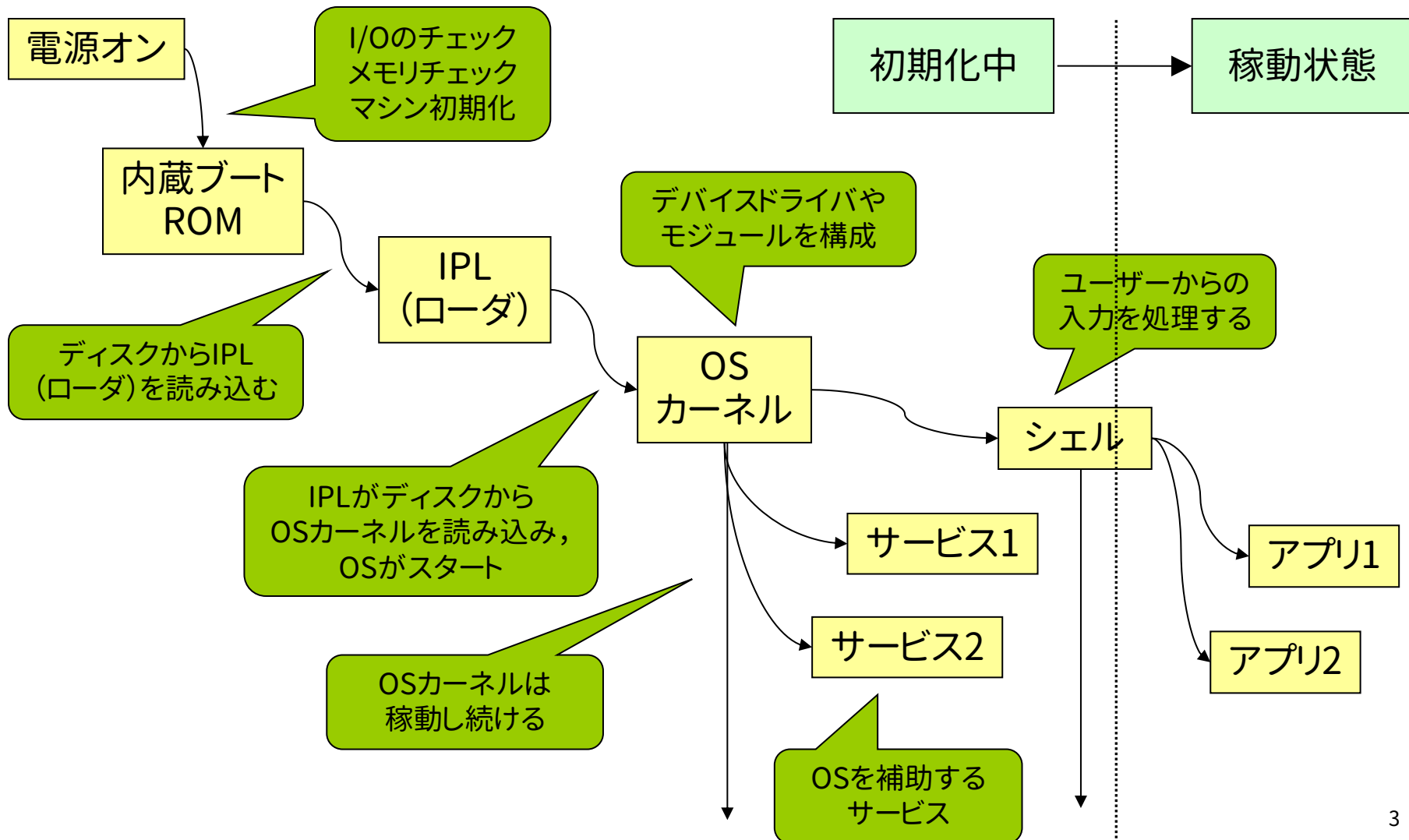
カーネルとデバイスドライバ

2020-03

# OSの構成要素



# OSの開始処理(ブートシーケンス)



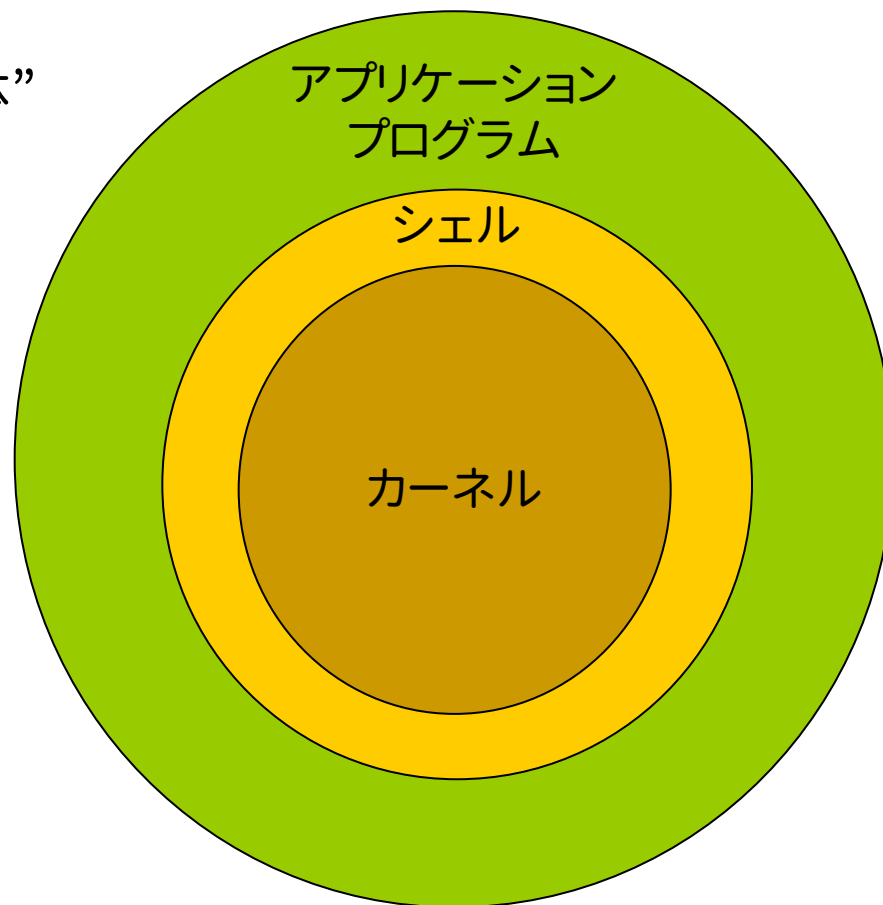
# Windowsのブートログ

---

- Windows 7まで
  - PC起動時に[F8]キーを押す
  - [ブートのログ作成を有効にする]を選択して,Windowsを起動
- Windows 8
  - コマンドプロンプトを起動
  - 「shutdown /r /o /t 0」と入力して,Windowsを再起動
- Windows 10
  - 「msconfig」(システム構成)というプログラムを検索し,実行
  - [ブート]→[ブート ログ]を選択して,Windowsを再起動
- C:¥Windowsフォルダに「ntbtlog.txt」というログが残る
  - 最初に読み込まれている ntoskrnl.exe がカーネル(約8Mバイト)
  - その後,各種のデバイスドライバ等が読み込まれているのが分かる

# カーネル

- Kernel=“核”
  - OSの中核=いわゆる“OS本体”
- OSの基本機能を提供
  - 資源の管理
  - システムの制御
- 例) Linuxカーネル
  - 実行時サイズ 数メガバイト
  - ソースコード 2100万行以上
  - <https://www.kernel.org>
- 例) Windowsカーネル
  - system32\ntoskrnl.exe
  - ソースコード 1700万行以上? (Windows全体では数千万行?)



# カーネルの役割(1)

---

- プロセス管理
  - プロセス=実行中のプログラムのこと
  - プロセスの生成(起動)や消滅(終了)の処理をする
  
- プロセスのスケジューリング
  - 複数のプロセスのCPUの利用時間の割り振りをする
  - プロセスやカーネル自身に, CPU使用権を順に割り当てる
  
- プロセスの同期 / プロセス間通信
  - 複数のプロセスが資源をめぐって競合するときに調整する
  - プロセスとプロセスが通信するしくみを提供する
  
- ファイルシステム
  - HDDなどの補助記憶(ストレージ)に記録するデータを管理する

# カーネルの役割(2)

---

- 記憶管理(メモリ管理)
  - 主記憶(メインメモリ)を管理し,プロセスごとに割り当てる
  - 入出力制御と連携して,仮想記憶を実現する
  
- 入出力制御
  - 入出力機器(外部ディスク,キーボード,画面等)を制御する
  - ユーザに分かりやすいコンピュータの操作手段を用意する
  
- タイマー管理
  - 現在時刻の管理,経過時間の監視・通知などを行う
  
- 割り込み制御
  - ハードウェアからCPUへの“割り込み”通知機能を管理する
  - 入出力割り込み,タイマ割り込み,演算割り込みなどを処理する

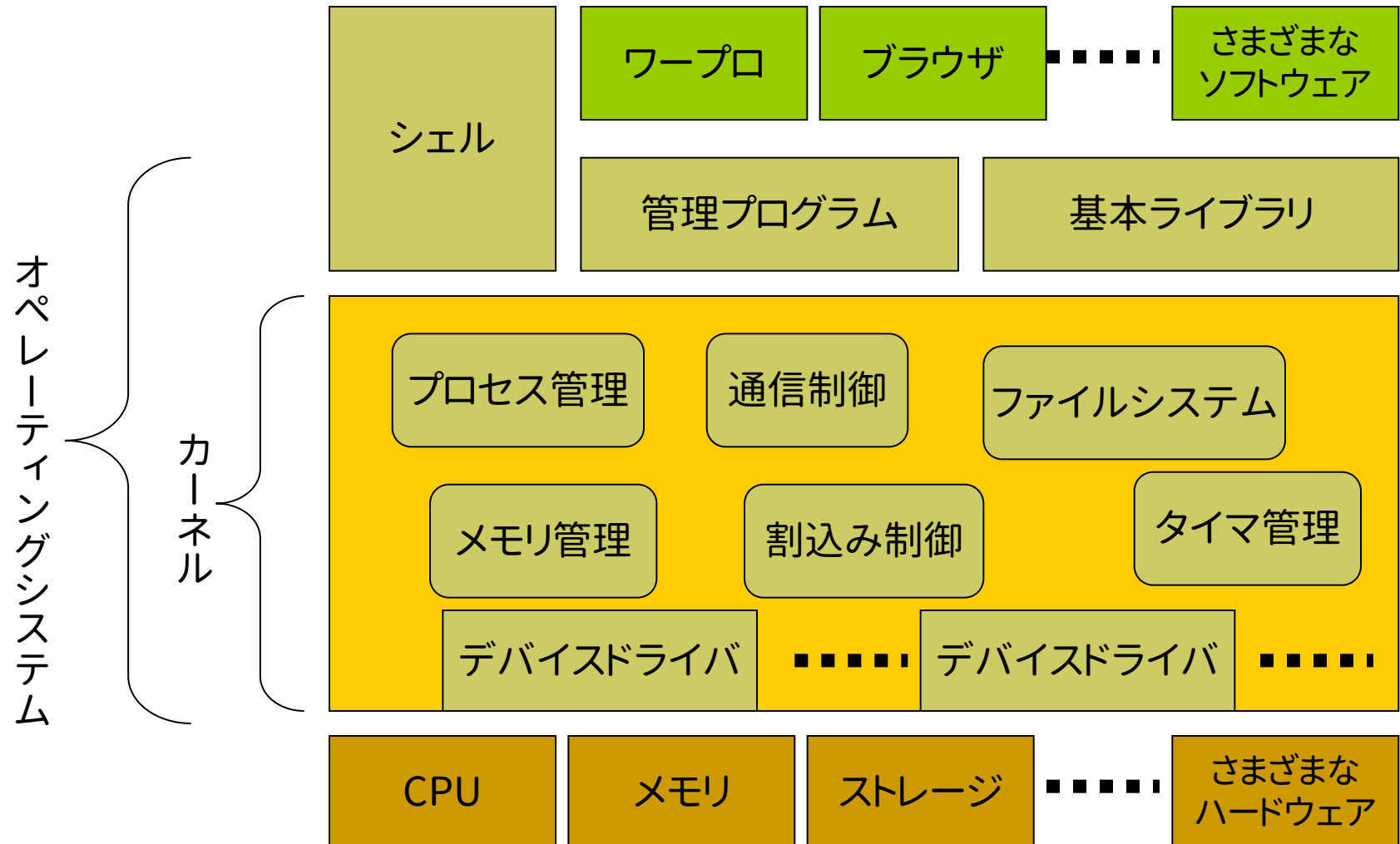
# OSの設計(アーキテクチャ)

---

- モノリシック(一枚岩)カーネル
  - OSのカーネルを,1つの巨大なプログラムとして開発する
  - 冗長性を抑え,実行速度が速いが,内部は入り組んだ設計になる
  
- マイクロカーネル
  - カーネル本体は,できるだけ小さく最小限の機能にとどめる
  - 他の機能は,別々の小さなプログラム(サービス)として開発する
  - 拡張・改良が容易で運用性も高まるが,実行速度が犠牲になる
  
- 中間的な手法
  - モジュール化: カーネルの機能をモジュール(部品)に分割しておき, OS起動時の初期化段階で,モノリシックカーネルのように結合する
  - デーモン / サービス: 通常のプログラムを常駐させることで, OSに機能を追加する (UNIX: daemon, Windows: service)



# OSの構成要素



# デバイスドライバ

---

- デバイス=ハードウェア装置
  - ハードディスクコントローラ, ネットワークアダプタ(LAN),
  - プリンタ, 内蔵時計, キーボード, マウス, ディスプレイなど...
  - OS内部には, デバイスごとの制御方式を組み込まないといけない
  
- デバイスドライバとは?
  - デバイスを駆動(ドライブ)するための専用のソフトウェア
  - カーネルの中の各デバイス(ハードウェア)に依存した部分のこと
  
- デバイスドライバのモジュール化
  - 昔は, カーネル本体にデバイスを制御するプログラムが埋め込まれ, ハードウェア構成の変更時には, OSの再コンパイルが必要だった
  - 現在は, カーネルに着脱できる「モジュール」(部品)になっている

# Windowsの例

---

## □ Windowsのサービス一覧

- サービス(常駐しているプログラム)の一覧を表示する
- Win8まで: [コントロールパネル]→[管理ツール]→[サービス]
- Win10: 田メニュー→[Windows管理ツール]→[コンピューターの管理]→[サービスとアプリケーション]→[サービス]

## □ Windowsのデバイスドライバ

- デバイスドライバの一覧を表示する
- Win8まで: [コントロールパネル]→[システム]→[ハードウェア]→[デバイスマネージャー]
- Win10: 田メニュー→[Windows管理ツール]→[コンピューターの管理]→[システム ツール]→[デバイス マネージャー]

# コンピュータハードウェア

## □ 制御機能/演算機能

- CPU (プロセッサ)

## □ 記憶機能

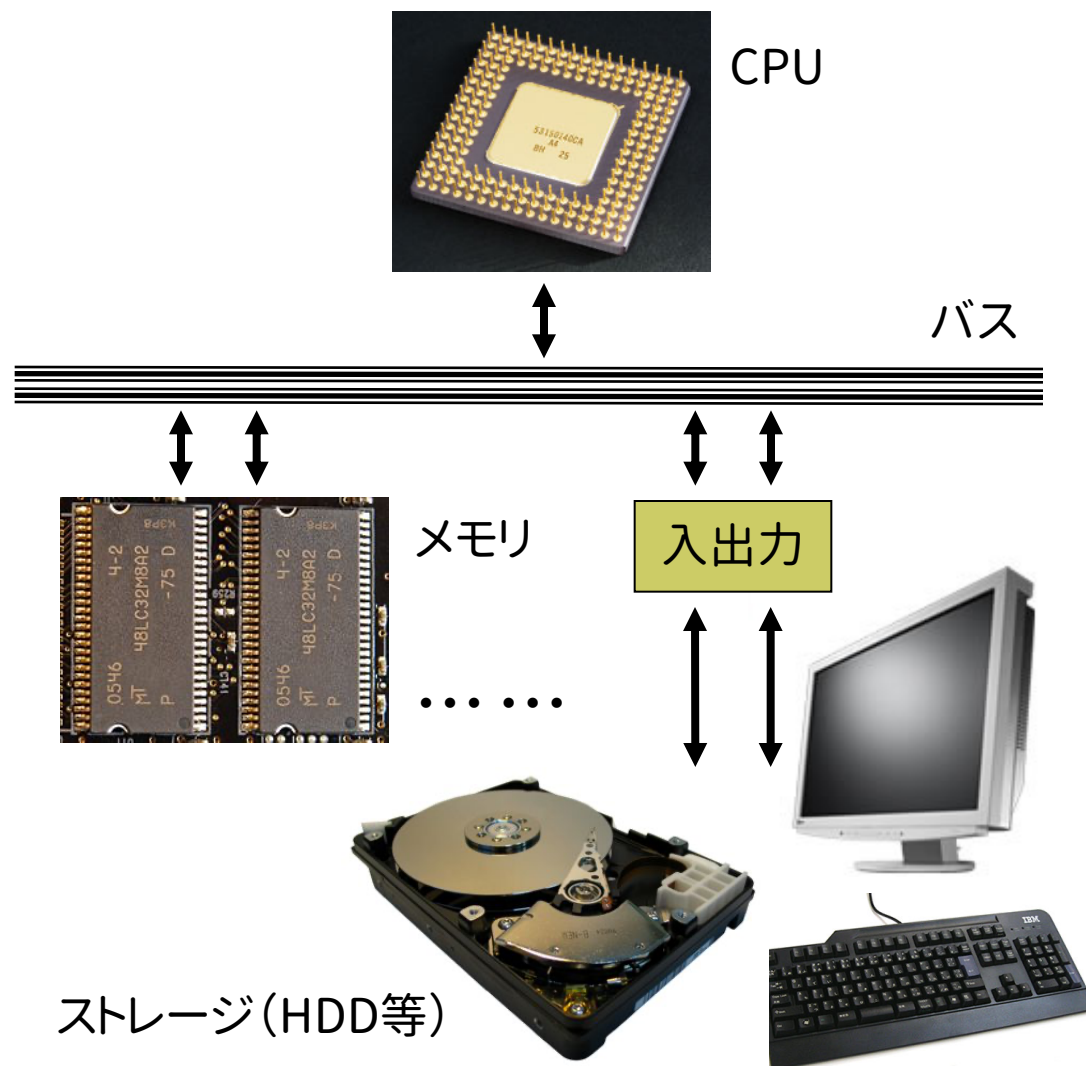
- 主記憶 (メモリ)
- 補助記憶 (HDD等)

## □ 入力/出力機能

- 各種周辺機器
- I/Oとも呼ばれる



デバイスドライバで制御



# デバイスドライバの動作方式

---

- デバイスとソフトウェアの関係
  - デバイス ⇔ OS (デバイスドライバ) ⇔ ソフトウェア
  - OSがデバイスドライバを駆動するケースは以下の2つ
  
- ソフトウェア側からの要求
  - ソフトウェアが, OSが提供するAPIを介して, デバイスを利用する
  - ソフトウェア → (API) → カーネル → ドライバ → デバイス
  - 例) ワープロによるファイルの読み書き, 音楽ソフトによる再生
  
- デバイス側からの要求
  - デバイスが, 「割り込み」などによってOSに対応を要求すると, OSがデバイスドライバを使って処理を行い, ソフトウェアに通知する
  - デバイス → (割り込み) → カーネル → ドライバ / ソフトウェア
  - 例) ネットワークからのデータ着信, センサーからの割り込み

# 「割り込み」?

---

## □ 割り込みとは

- ハードウェアなどから、処理要求がCPUに通知されるしくみ
- ハードウェア的な機能として実現されている(一種の電子回路)
- 通常のOSでは、割り込みはOSのカーネルによって処理される

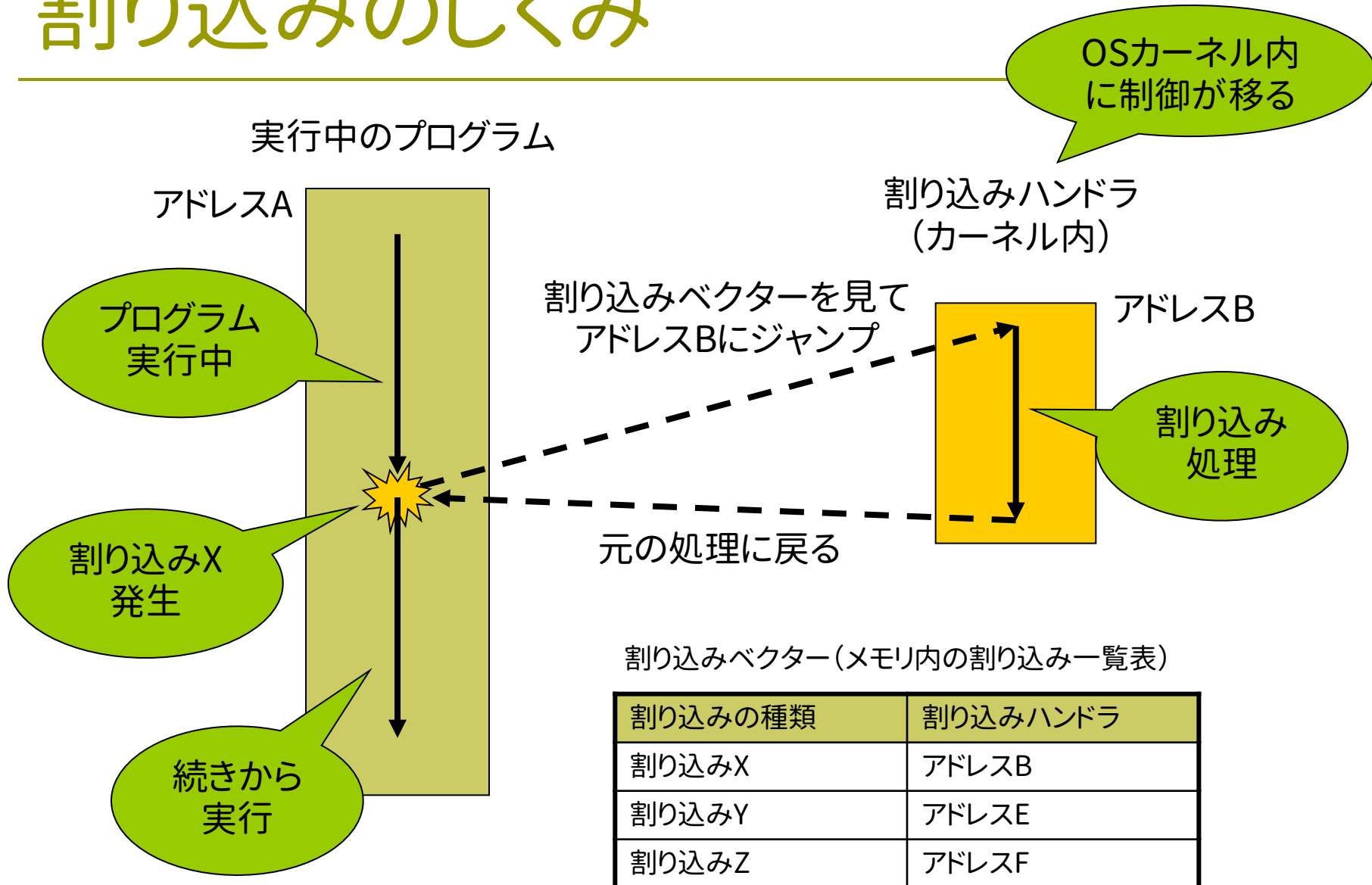
## □ 外部割り込み

- ハードウェアの状態変化で発生する割り込み
- 入出力割り込み: ストレージ(HDD等)からの読み込み完了で発生
- タイマー割り込み: 設定時間の経過で発生

## □ 内部割り込み(例外,トラップ)

- CPU内部の演算・処理等で発生する割り込み
- ゼロ除算割り込み: ゼロによる割り算で発生
- SVC割り込み: 故意の割り込みでOSに制御を渡し,APIを実行

# 割り込みのしくみ



# 割り込み処理(例外処理)

---

## □ 割り込み処理

- 割り込みが発生すると,CPUは実行中の処理(プロセス)を中断し,登録されている「割り込みハンドラ」を優先して実行する
- 中断されたプロセスは,割り込まれていることには気づかない
- 割り込みハンドラが終了すると,中断した処理に戻って続行する

## □ 割り込みハンドラ

- 割り込みの発生時に作動するように登録してあるプログラム
- 通常は,割り込みの種類ごとにOSがカーネル内に用意しておく

## □ 割り込みベクター

- 割り込みの種類と割り込みハンドラの対応表
- CPUが主記憶に専用の領域を設定しており,OSはそこを書き換える



# 演習課題

---

- Webフォームで入力
  - BB(メール)参照
  
- 実習課題はお休み
  - 次回以降,実習課題が増えます。