

# オペレーティングシステム

樋口さぶろお

龍谷大学工学部数理情報学科

情報処理の基礎 L12(2015-01-07 Wed)

## 今日の目標

- オペレーティングシステムの例を挙げられる
- オペレーティングシステムの機能を説明できる
- コンピュータが起動してから Word と Excel が同時に起動するまでに起きることを (一定の詳しさで) 説明できる



<http://hig3.net>

## L11-S1

## Quiz 略解+コメント:機械語

## 略解

- 読込 [A][1200]
- 読込 [B][1204]
- 演算 (加算)[A][B]
- 書出 [A][1214]
- 読込 [A][1208]
- 読込 [B][120C]
- 演算 (減算)[A][B]
- 演算 (減算)[A][B]
- 読込 [B][1214]
- 演算 (乗算)[A][B]
- 書出 [A][1210]

## ここまで来たよ

- 1 略解:アセンブリ言語と機械語
- 2 オペレーティングシステム
  - オペレーティングシステムとは
  - OS によるモジュール化
  - OS によるマルチタスク化
  - マルチタスクの管理

## オペレーティングシステムとは

現在の一般的なコンピュータの構成

$$\begin{aligned}\text{計算機システム} &= \text{ハードウェア} + \text{ソフトウェア} \\ &= \text{ハードウェア} + (\text{OS} + \text{AP})\end{aligned}$$

- **OS**=Operating System オペレーティングシステム
- **AP**=Application Program アプリケーションプログラム

ハードウェアコンピュータの機械としての実体.

語源:

種類の例: PC, Mac, Android タブレット, iPad, ...

**AP** それ自体, 人間にとって意味のある入出力をするプログラム

例: ワードプロセッサ, Web ブラウザ, エディタ, ...

## オペレーティングシステムの目的と機能

オペレーティングシステム AP の実行を助け、調整する基本ソフトウェア

### オペレーティングシステムの目的

- ソフトウェアのモジュール化: ソフトウェアの開発と保守を楽にする
- 計算機システムのマルチタスク化: 複数の AP を独立/協調して実行させる

その方法:  = virtualization

AP が、ハードウェアの機能をそのまま使わず、ハードウェアのふるまいを真似るソフトウェア (=いまは OS) を通して利用するようにする。

### オペレーティングシステムの例

- Windows (PC)
- Linux (PC, いろいろ) Ubuntu, Redhat, Fedora, SuSE, ...
- Unix (PC, いろいろ) Solaris, FreeBSD, ...
- OS X (Mac)

## ここまで来たよ

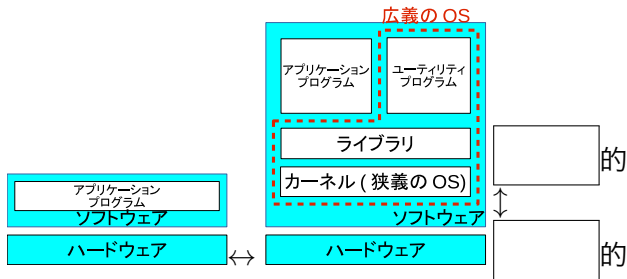
### 1 略解:アセンブリ言語と機械語

### 2 オペレーティングシステム

- オペレーティングシステムとは
- OS によるモジュール化
- OS によるマルチタスク化
- マルチタスクの管理

## OS によるモジュール化

これまで語ってた計算機システム ↔ OS のある実際の計算機システム



上のレイヤー (層) が, 下のレイヤーの機能を利用している. 上が下に依存している.

仮想化=あるレイヤー (例 AP) の下にソフトウェアのレイヤーをはさむ.  
AP の直下をハードウェアからソフトウェアに変更する.

## ライブラリ

- 多くの AP が共通に持ちたい機能がある. 各 AP が持つと開発も配布もたいへん
- ハードウェアよりのプログラム (例: キーボード読取プログラム) はハードウェアごとに異なる. 全ソフトウェアメーカーにキーボード読取プログラムを送りつける? 新しいキーボードが発売されるたびに更新?

→ そういう部分は OS に任せ, 各 AP は OS の機能呼び出す.

OS 側の, サービス受付窓口:  = Application Programming Interface  
似た概念: ライブラリ, システムコール

### ライブラリの例

printf, scanf, ファイル保存ダイアログ, コピーバッファ

### ユーティリティプログラム

ユーザが直接使うが, 計算機システムの保守, 調整に使うもの. 例:

Windows Update.



## Quiz(オペレーティングシステム)

次のうちオペレーティングシステム (OS) についてうそはどれ?

- ① オペレーティングシステムはプログラミング言語で記述されたプログラムである
- ② アプリケーションプログラムの種類のほうが、OS の種類よりも多い
- ③ アプリケーションプログラムよりも OS のほうがハードウェアに近い
- ④ 同じ PC なら、OS が異なっても、同じアプリケーションプログラムがそのまま動作する

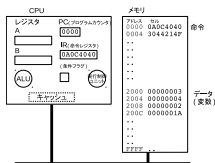
## ここまで来たよ

### 1 略解:アセンブリ言語と機械語

### 2 オペレーティングシステム

- オペレーティングシステムとは
- OS によるモジュール化
- OS によるマルチタスク化
- マルチタスクの管理

## OS がない計算機システムの起動から終了まで



- ① 何らかの方法で人間がメモリに 1 個の AP のプログラムを書き込んでおく
- ② 人間が電源オン
- ③ PC=0000 から実行開始
- ④ AP のプログラムが演算や入出力を行う
- ⑤ プログラム内の停止命令で停止
- ⑥ 人間が電源オフ

レストランの例 レストランの調理場に例えると…

## こうだと便利なのに…

- 人間の気まぐれに次々に AP を実行したい ハードウェアよりソフトウェアのほうが楽
- 複数の AP を同時に実行したい マルチタスク
- 複数の AP を協調させて動作させたい 例えば



- 複数のユーザが同時に 1 個のコンピュータを使いたい

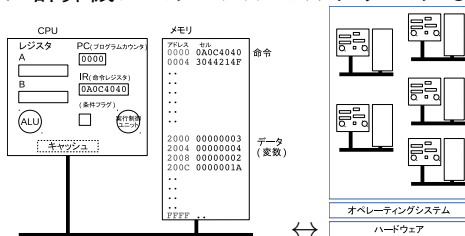
Windows や Linux では実際にそういうことが可能になっている。  
現在実行されているプロセスを観察する方法 → 次週

- Windows タスクマネージャー Control+Alt+Delete
- Linux ターミナルで `ps aux`
- iOS ホームボタンダブルクリック (AP のみが表示)

## OS のあるコンピュータの起動

- ① 0000-1000(例) のメモリは ROM. ここにブートローダというプログラムが書き込まれている
- ② PC=0000 でブートローダが実行される.
- ③ ブートローダは, 補助記憶装置 (ハードディスク) からメモリ (RAM) の 1000(例) 以降に OS というプログラムを読み込む
- ④ 分岐命令により PC が 1000 を指す
- ⑤ OS はユーザの入力をうけとる.
- ⑥ ユーザが AP の実行を指示するたびに, AP のプログラムが補助記憶装置 (ハードディスク) からメモリ (RAM) に読み込まれ, CPU-メモリをエミュレートするプロセスが作られて実行される

## これまで語ってた計算機システム ↔ マルチタスクな計算機システム



**ブートする** = 起動する 語源: bootstrap, ブートストラップ(くつひも)

**エミュレートする** = 模倣する

**プロセス** = プロセッサに処理されるもの. CPU = Central Processing Unit.

レストランの例

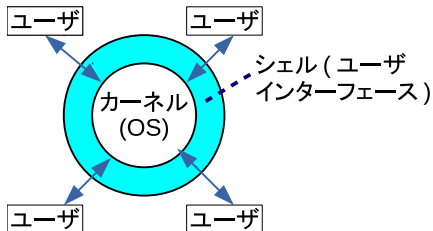
OS = キッチンマネージャー

複数の注文がいろんなタイミングで来たときに、料理人に割り振り、コンロ、材料、水道の利用を管理し、味付けが混ざらないようにする。

## シェル

### 実行の指示を受け取るユーティリティプログラム

- Linux ならターミナルの中で動いているプログラム. 指示方法:  
./a.out
- Windows ならエクスプローラやタスクマネージャ. 指示方法:ダブル  
クリックやファイル名指定.
- iOS ならホーム画面. 指示方法:アイコンタッチ



シェルは OS の皮膚にすぎない. 語源:

## Quiz(マルチタスキング)

次のうち、OS のマルチタスキング機能を活用しているわけじゃないのはどれ?

- ① Word と Excel を同時に開いてコピーアンドペースト
- ② プリンタに印刷している間もワープロで編集ができる
- ③ シューティングゲームで敵機が動いている間も自機は動ける
- ④ ワープロ使っているうちに自動的に Windows Update されてた



## ここまで来たよ

① 略解:アセンブリ言語と機械語

② オペレーティングシステム

- オペレーティングシステムとは
- OS によるモジュール化
- OS によるマルチタスク化
- マルチタスクの管理

## CPU の管理=プロセス管理

本当の CPU は 1 個しかないのにどうやって複数プロセス?

両方のプログラムをメモリにおき, **時分割=タイムシェアリング**方式で実行.

OS が管理して, 一定時間 (例: $10^{-3}$ s) ごとにタスク A,B,C,... の間を切り替える.

タスク切替=PC を適切な値におき, レジスタの値を復元する.

**レストランの例** 複数の料理を同時に作りたいのにもコンロは 1 個しかない...

## メモリの管理

実記憶 (物理的) ↔ 仮想記憶 (論理的)

プロセスからはアドレス連番のひとつながりのメモリ (仮想記憶) が見えるが, 実際には, ページ単位で, メモリ (実記憶) 上のいろいろな場所や, ハードディスクに置かれ, OS がアドレスを変換する.

あるプロセスが, 別のプロセスの仮想記憶にアクセスできないように OS が監視している.

**ページング** 実記憶が不足したとき, 一部を補助記憶装置 (ハードディスク) に置くこと

**ページアウト** 当面不要な情報をメモリからハードディスクに追い出す

**ページイン** 必要に応じてメモリに戻す

速度と価格のトレードオフ: メモリはハードディスクより速いが, 高価  
**レストランの例** 調理台は狭い…

## Quiz(メモリ管理)

次のうち、計算機のメモリを増設しても解消しなそうなものは？

- ① 動画サイトの再生が始まるまでが遅い
- ② ゲームを5個までは同時にできるけど、6個目からは遅くなる
- ③ 長い動画で、解像度を変更する編集に時間がかかる
- ④ 表計算しているときに、最小化していたワープロを呼び出すのに時間がかかる

## デバイスの共有・管理

2種類のワープロから同時に「印刷」しても、文書が混ざって印刷されることはない

OSが、プリンタを管理していて、一度には1個のプロセスに許可を与える。**排他処理**

他のプロセスの印刷要求は待機場所に貯められる (**スプール**される)

**デッドロック**の危険

**レストランの例** 複数の料理を作りたいのに炊飯器は1個しかない…

**レストランの例** 複数の料理を作りたいのに2人の料理人が炊飯器と圧力鍋をとりあった…

## 補助記憶装置 (ハードディスク) の管理

2種類のワープロから‘同時’に「保存」しても、文書が混ざって保存されることはない

ファイルという単位で制御されることが多い。

出入はタイムシェアリング的に。

レストランの例 複数の料理を作りたいのに冷蔵庫は1個しかない…



## Quiz(OS の機能)

次のうち、OS の機能と関係してなさそうなのはどれ?

- ① C の課題で、間抜けなプログラムを書いて実行しても、Web ブラウザやエディタが悪影響を受けることはない
- ② C の課題で、無限ループを持つプログラムを書いて実行しても、Control-C で止めることができる
- ③ C の課題で、ターミナルから自分のプログラムを実行できる
- ④ C の課題で、どのコンピュータでも同じプログラムを実行すると同じ結果になる



## 連絡

- 2015-01-14 水 3 たぶん直前に教室変更. 掲示/ポータル/メールに注意.
- 2015-01-14 水 54 たぶん **プチテストリベンジ** 希望者のみ参加. **参加登録受付中**. プチテストと同じ出題計画. これと前回のプチテストのうち, 点数の高いほうを 30 ピーナッツ分とします. プチテストで 70% 以上を得た人は, 無視してファイナルトライアルに集中することをおすすめします.
- 2015-01-15 木 ころまで. **レポート問題提出期限** RaMMoodle から個人別問題取得.
- 2015-01-21 水 たぶん補講
- 2015-01-28 水 3 たぶんファイナルトライアル (40 ピーナッツ) **別紙参照**.
- 樋口のオフィスアワー 木 6(1-539), 金 昼 (7-002/1-502).
- チューターは **月**火水木 昼 (1-614).