

Operating System 2

岩井将行

資料ダウンロード先

<http://www.cps.im.dendai.ac.jp/index.php?Classes%2F2014OS>

講師

- 広島県福山市出身
 - 慶應義塾大学 卒
 - -2012東京大学生産技術研究所 助教
 - 2013-東京電機大学未来科学部情報メディア学科 准教授
-
- Facebook /masa.iwai
 - Twitter @masaiwai
 - iwai@im.dendai.ac.jp

TA・SA・副手

- Shige
- 実空間コンピューティング研究室
- 東京電機大学大学院 未来科学研究科情報メディア専攻
- shige@cps.im.dendai.ac.jp

OS概論

- オペレーティングシステム (Operating System、OS) はCPUやメモリ、周辺機器であるキーボードやディスプレイなどを管理して、ユーザにそれらデバイスへのインタフェースを提供しています。
- OSの基礎を理解し、コンピュータやSmartPhoneの基本動作原理を理解ことを目標とします。さらにサーバ管理やVMについて学びます。

最新の身近なOS

- Android/Dalvik VM
- Linux Ubuntu
- iOS
- MacOSX Lion/Mountain Lion
- WindowsPhone7.8
- Windows8

目標 1

- 以下のOSに関わる基礎的な用語を図や文章・コマンドを使い説明できるようになることを目標します。
- CPU、デバイス、割り込み、記憶装置、ハードディスク装置、RAID、パリティ、スレッドとプロセス、メモリアドレス空間、ファイルシステム、NIC、ソケット、カーネル、ログイン、コマンド操作、シェル
- プロセス管理、時分割処理とプロセス切り替え、スケジューリング、プロセス表、メモリ階層、キャッシュ、アドレス空間、物理アドレスと論理アドレス、ページング、チェックポイント、効率的な自動メモリ管理、GC、シェルスクリプト、ファイルのバックアップ、アクセス権、ドライブ、ディレクトリ、ファイル、open/read/write
- ファイルのメモリへのマッピング、アクセス制御、権限、空き領域管理、
- ethernet,ping,socket,tcp/udp,rpc,apach,http,ssh,ftp、
- Android, iOS, windows8,maxOS X,VM

目標2

- **基本情報技術者試験（FE）**
- **応用情報技術者試験（AP）**
- **のどちらかを受験することを意識させる。**
- <http://www.jitec.ipa.go.jp/>

採点方法

- 出席点 20点満点
- 毎回のミニテストかレポート平均 20点満点
- 最終試験 60点満点
 - 中間試験がある場合(中間20点 最終40点満点)
- それぞれ数値化し
- 絶対評価

出席

- 10分前から開始までに一応タッチしてください。
- 代返を防ぐためにTAが人数カウントします。
- 運用は適宜見直します。

第1回

- ハードウェアとOS
- CPUとデバイス、割り込み、記憶装置、ハードディスク装置、RAID、パリティ
- スレッドとプロセス、メモリアドレス空間、ファイルシステム、NIC、ソケット、カーネル

第2回

- CUI
- タイピング, ログイン, コマンド操作, マニュアル, シェル
- ファイル操作, エディタ

第3回

- プロセス, ジョブ
- プロセス管理、時分割処理とプロセス切り替え、スケジューリング、プロセス表
- 東京電機大学 学生ポータルサイト 3
<http://portal.sa.dendai.ac.jp/up/faces/up/km/pKms0804B.jsp?nend...>
- 1

第4回

- 記憶装置
- メモリ階層、キャッシュ、アドレス空間、物理アドレスと論理アドレス、ページング、チェックポイント
- グ、効率的な自動メモリ管理、GC、フラッシュメモリ、HDD、SSD

第5回

- シェルとアクセス権
- 標準入出力、フィルタコマンド、シェルスクリプト、ファイルのバックアップ、アクセス権、ドライブ、ディレク
- トリ、ファイル、open/read/write
- ファイルのメモリへのマッピング、アクセス制御、権限、空き領域管理

第6回

- ネットワークとOS
- ethernet,ping,socket,tcp/udp,rpc,apach,http,ssh,ftp,remotewindow
- **最新のOS事情**

第7回

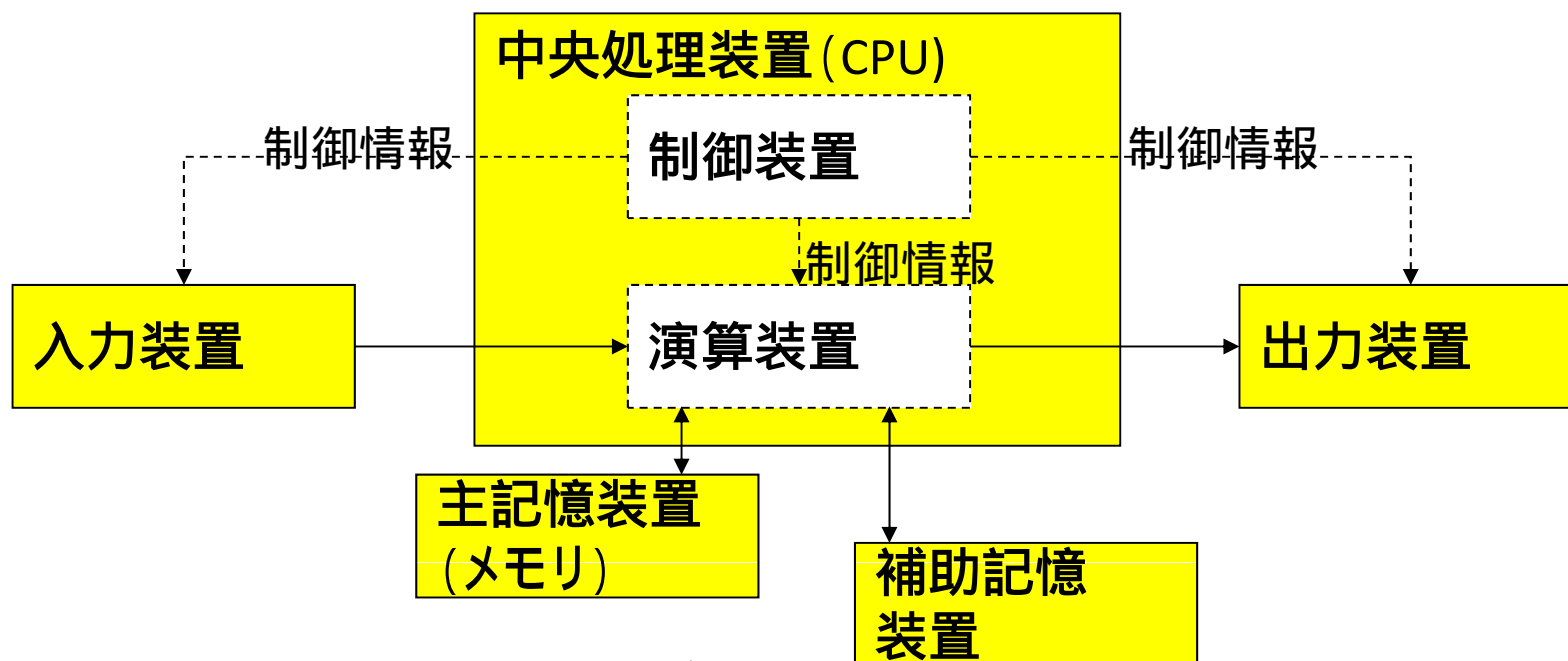
- 試験

前回の復習

ハードウェアとOSについて

ハードウェアの機能と構成

- コンピュータは、制御・演算・記憶・入力・出力の機能を持つ
- 以下のユニットで構成される



ハードウェアの概観と機能

CPU 2

- クロック
 - CPUの各部分の基本動作のタイミングを合わせるために、一定の時間間隔で発生している信号
 - CPUが行う様々な処理はクロック信号にタイミングを合わせて行う
 - クロック信号が発生する間隔が短ければ短いほど、CPUの動作が高速
- クロック周波数(Hz)
 - クロック信号が1秒間に何回発生するかを示す数値
 - 一般にその周波数が高ければCPUの動作速度が速い

ハードウェアの概観と機能

主記憶装置

- データの読み込み速度・書き込み速度が(補助記憶装置と比べて)速い
- 電源を切ると記憶されていたデータは消えてしまう



メモリスロット



メモリの本体

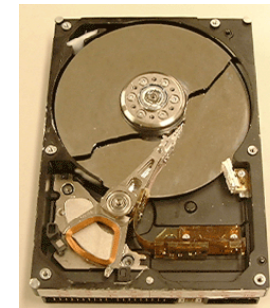
ハードウェアの概観と機能

補助記憶装置 1

- ハードディスク
 - 回転する円盤に磁気を帯びさせる事で、情報を保存
 - 一連の情報は同心円状に配置されたトラック(円周)に記録される
 - ヘッド(銀色の三角形の形をした突起の先)が円盤の上を移動することで情報を読み書きするトラックを変える



ハードディスク
概観



ハードディスク
内部

ソリッドステートドライブの利点と欠点

- 利点

- シークタイムがないためランダムアクセス性能に優れる
- 物理的な稼動箇所がないため省電力、動作音がしないので静か
- 同じ理由でHDDよりはるかに振動・衝撃に強い

- 欠点

- 容量単位の価格がHDDより高い(2012年現在、HDDが1GBあたり5～10円に比べSSDは100円前後)
- 書き換える度にトンネル酸化膜が確実に劣化するため、24時間常時書き込みを行うような用途の場合HDDより寿命が短くなる場合がある

ハードウェアの概観と機能

補助記憶装置 2

- その他の補助記憶装置

- フロッピーディスク(もうない)



- USBフラッシュメモリ SDメモリ

- CD-ROM / CD-R / CD-RW

- Blu-ray Disc/DVD-R

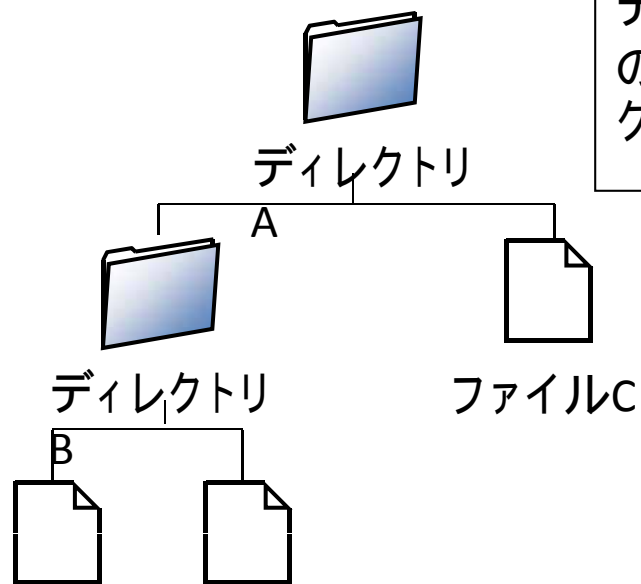
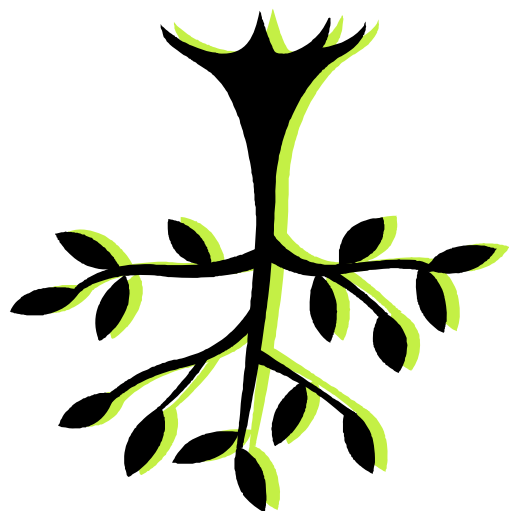


DVDドライブ

ファイルとディレクトリのツリー構造

path

- ディレクトリの中にディレクトリを入れることができる
- 図に描くと、枝分かれした(逆さまの)木のような構造(ツリー構造)になる



ディレクトリAをディレクトリBの親ディレクトリ
ディレクトリBをディレクトリAの子ディレクトリ、サブディレクトリと呼ぶ

OSの歴史

OSの始まり

- 一番最初のコンピュータ(1950年代～)
 - パンチ(穴)カードを使って磁気テープに読み込む
 - 読み込まれたデータの実行が主な仕事
 - プロセス状態の管理
 - コンピュータの機種ごとに、専用のOS
- 1960年代前半のOS:
 - 1950年代の機能 + α (いろいろ:略)
- 1960年代後半: 大型コンピュータ
 - 様々な機種が発売された
 - それら多くの機種に、1つのOSで対応した



↑ IBM System/360 (1964)

文字の世界

- 1970 ~ 80年代前半：文字ベースのシステム

- UNIX の始まり

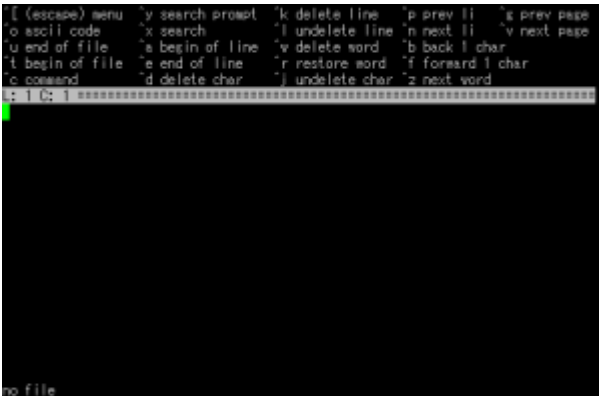
この時代のテキストエディタ
(ee) は今でも使える →

- コマンドライン
 - テキストエディタ、文章の清書

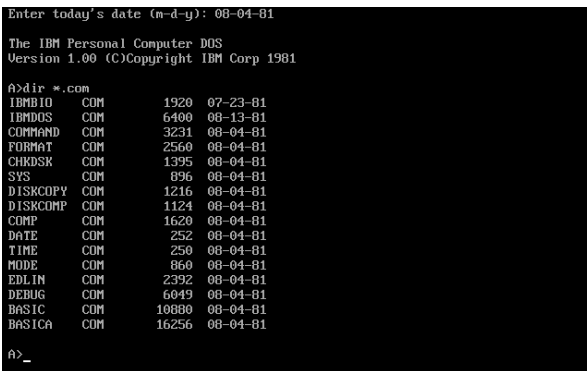
- DOS (Disk Operating System)

- IBM のコンピュータ用OS：IBMの多くのシステムで使われた
 - MS-DOS：Microsoft 製：Microsoft が有名になり始める
 - 他、IBM PC-DOS など
 - Apple DOS：Apple Computer 製

IBM PC-DOS (1981) →



```
[ (escape) menu      y search prompt    k delete line      p prev ll          g prev page
o ascii code        x search           l undelete line    n next ll          v next page
u end of file        a begin of line    w delete word      b back 1 char
t begin of file      e end of line      r restore word     f forward 1 char
c command            d delete char      j undelete char    z next word
```

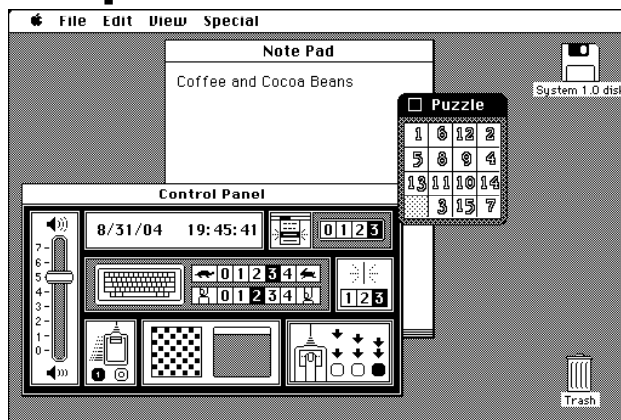


```
Enter today's date (m-d-y): 08-04-81
The IBM Personal Computer DOS
Version 1.00 (C)Copyright IBM Corp 1981
A>dir *.com
DIR 10 1920 07-23-81
DIR 10 5400 08-13-81
DIR 10 3231 08-04-81
DIR 10 2560 08-04-81
DIR 10 1395 08-04-81
DIR 10 896 08-04-81
DIR 10 1216 08-04-81
DIR 10 1124 08-04-81
DIR 10 1620 08-04-81
DIR 10 252 08-04-81
DIR 10 250 08-04-81
DIR 10 860 08-04-81
DIR 10 2392 08-04-81
DIR 10 6049 08-04-81
DIR 10 10800 08-04-81
DIR 10 16256 08-04-81
A>_
```

グラフィカルな世界

- 1980年代 ~ : GUI (Graphical User Interface)
 - WYSIWIG: What You See Is What I Get
- 1984年 : Mac OS System 1.0
 - マウス、アイコン、ウィンドウ、ゴミ箱、メニューバー
 - GUIでも画面は白黒
- 1985年 : Windows 1.0: **ウィンドウを重ねられない**

い



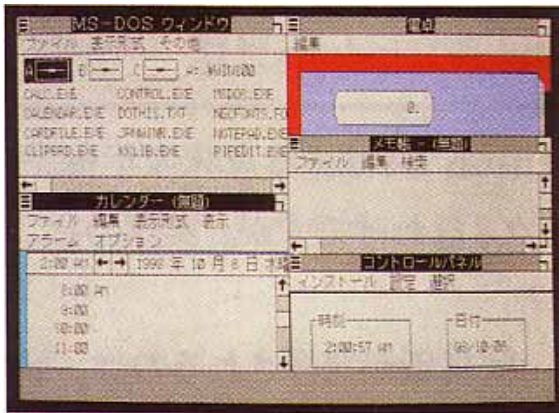
↑ Mac OS System 1.0



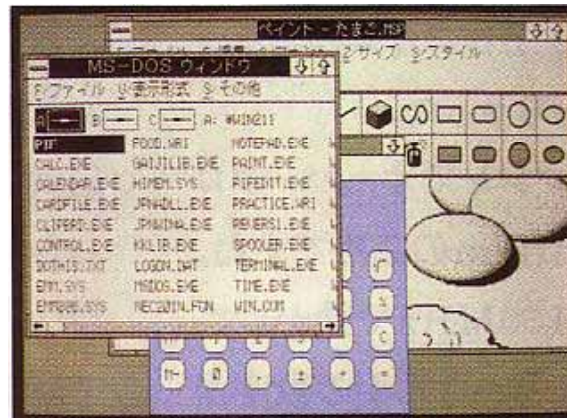
↑ Windows 1.0

少しずつ、グラフィカルに

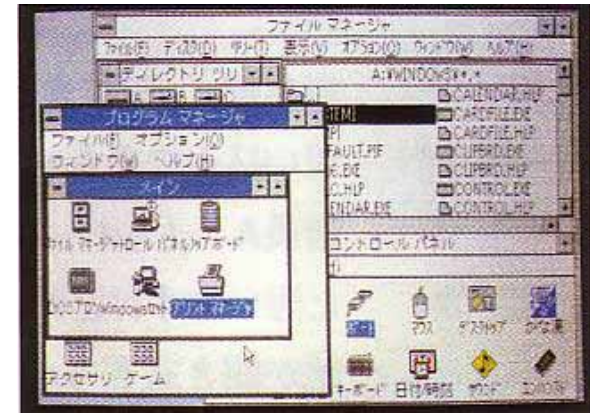
昔のWindows



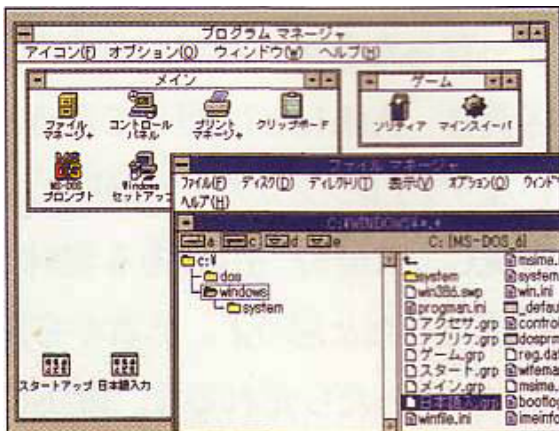
↑ Windows 1.0 (1985)



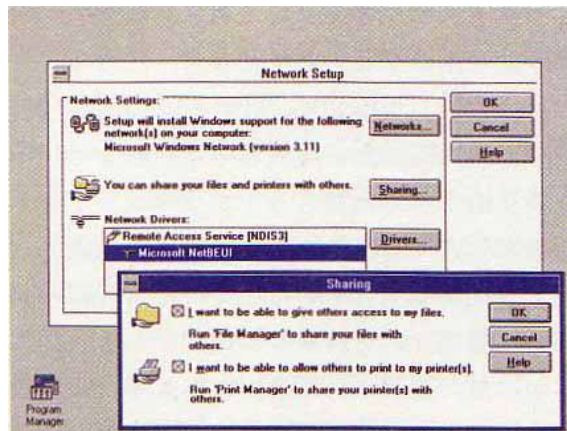
↑ Windows 2.11 (1988)



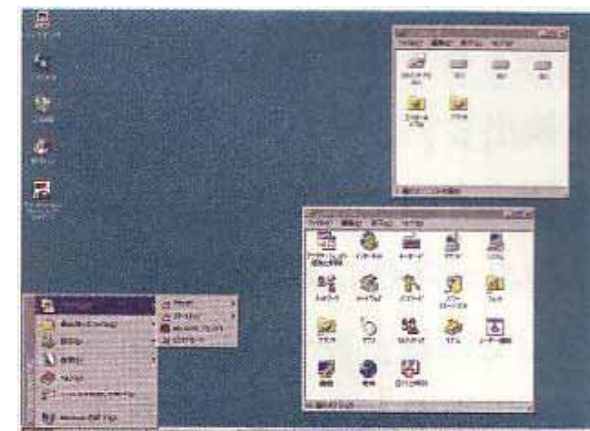
↑ Windows 3.0 (1990)



↑ Windows 3.1 (1992)



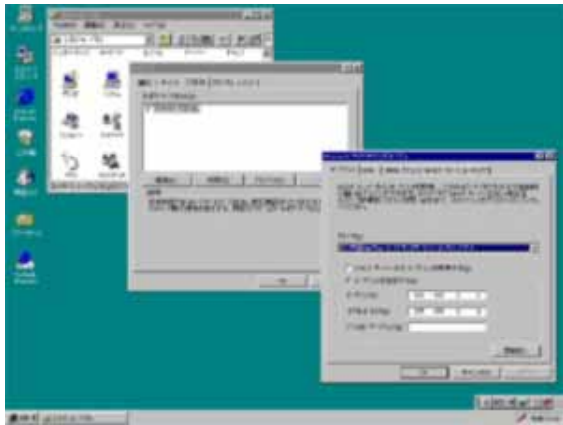
↑ Windows for Workgroups 3.11
オペレーティングシステム2014



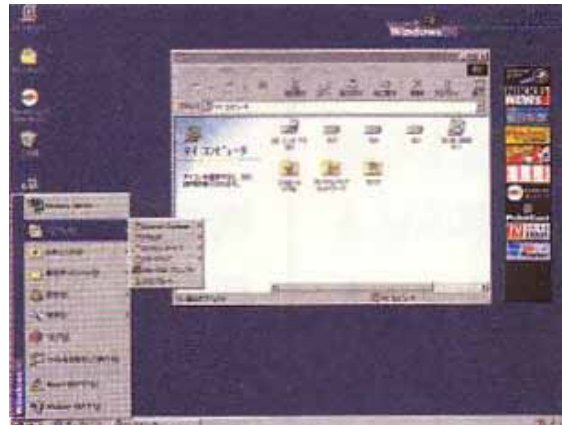
↑ Windows 95 (1995)

少しずつ、グラフィカルに

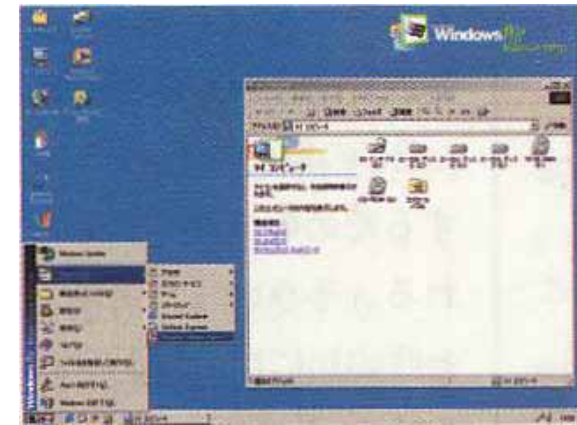
最近の Windows



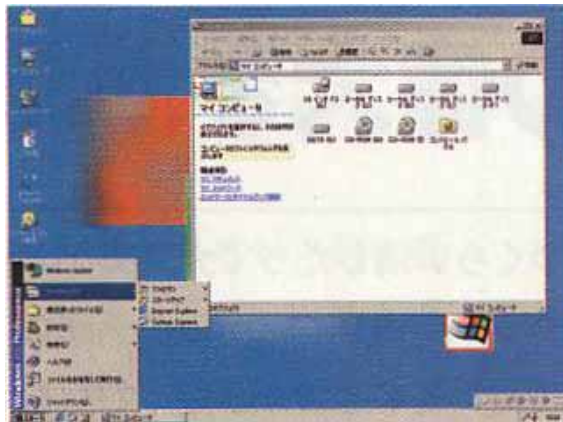
↑ Windows NT 4.0 (1996)



↑ Windows 98 (1998)



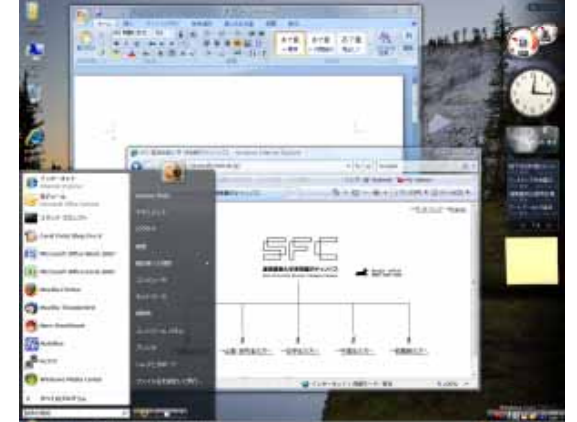
↑ Windows Me (2000)



↑ Windows 2000 (2000)



↑ Windows XP (2002)
オペレーティングシステム2014



↑ Windows Vista (2007)

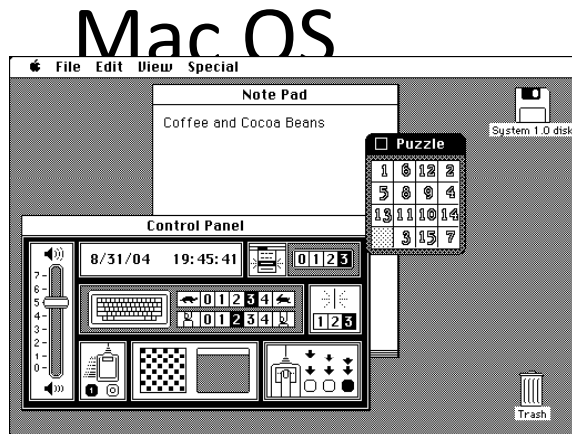
- <http://windows.microsoft.com/ja-jp/windows-8/meet>

Windows8/8.1 Modern UI

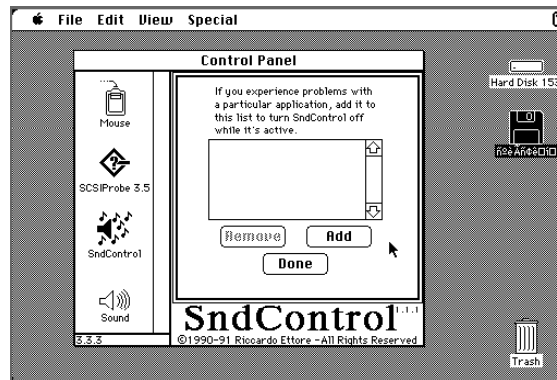


テン

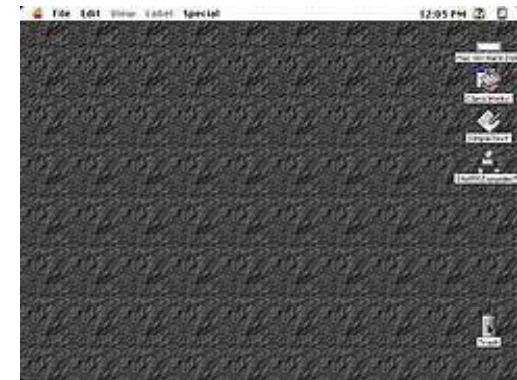
少しずつ、グラフィカルに



↑ Mac OS System 1.0 (1984)



↑ Mac OS System 6.0 (1989)



↑ Mac OS System 7.5 (1995)



↑ Mac OS 8 (1997)



↑ Mac OS 9 (1999)
オペレーティングシステム2014



↑ Mac OS X 10.4 (2005)

Mac OS X mountain lion


**OS X
Mountain Lion**
世界で最も先進的なコンピュータの
オペレーティングシステムが、さらに進化。
Mac, iPad, iPhoneの連携も、より優れたものに。
Mac App Storeで販売中、
1,700円



MacOS X mavericks

- <http://www.apple.com/jp/osx/whats-new/>

OS X Mavericksの新機能。

デスクトップ用の新しいアプリケーションの数々、使うのが楽しくなる様々な機能、そして消費電力を抑えながら、Macのパフォーマンスをさらに高めるためのテクノロジーを搭載。

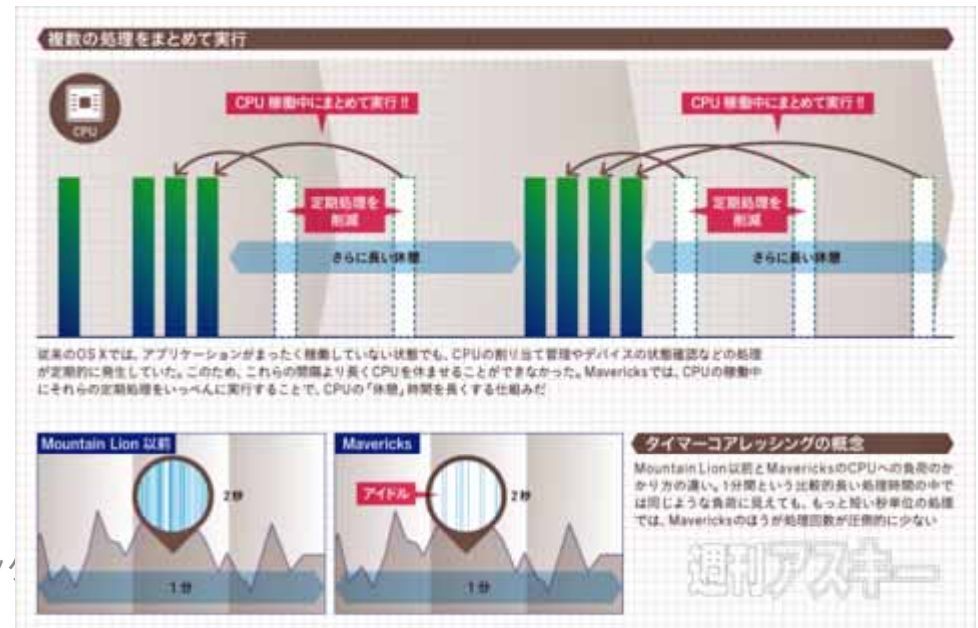
OS X Mavericksで、Macでの体験が一段と優れたものになります。



MacOS X mavericks

- タイマーコアレスシング
– 低負荷のプロセスをまとめて処理してアイドル時間を増やすことにより、CPUの電力消費を抑える。
- App Nap 他のウィンドウの背面に隠れているアプリの処理速度を低下させることにより、電力消費を抑える。
- 圧縮メモリ 非アクティブのアプリのメモリを自動的に圧縮するようになる。

- iBooks
– る。
- マップ
– iOS 6で追加されたアプリ。



OS の進化で出来るようになった事 (一部)

- **新しいハードウェアのサポート**
 - Windows 95 では、USBは使えない
- **ファイル名の文字数**
 - Windows 1.0 ~ 3.1: 8文字、95 ~ Me: 32文字、2000 ~ : 256文字、Mac OS ~ 9: 31文字、etc...
- **ユーザインタフェース:**
 - メニュー、見た目、操作性、etc...
- **様々な機能の統合**
 - ブラウザの統合、メディアプレーヤー、ムービーメーカー、システムの復元、アップデート機能、セキュリティ向上機能、デスクトップ検索、etc...

出来るようになった事 vs. 変わらない事

- さて、OSの進化と共に色々な機能が增加。
- では昔から今までのOSで、共通している事は？
 - OSの3つの仕事：
 - ハードウェアの抽象化
 - ユーザインタフェースの提供
 - 資源の管理
 - 新しいデバイスが使えるようになったり、インタフェースとして新しく変更された所は色々ある。
 - 最近の流行：セキュリティ対策、分かり易いGUI
 - でも、OSの仕事は、OSが変わっても根本的には同じ。

OSの役割

- OSの役割は、人間がコンピュータを使いやすくすること
- 「アプリケーションソフトとハードウェアの仲介」「ユーザインタフェースの提供」「資源の管理」という3つの役割がある

ユーザインタフェースの提供

- ユーザがコンピュータとやり取りを行う部分の総称がユーザインタフェース (UI)
- UIは、ユーザに対する情報の表示形式やユーザのデータ入力方式を規定する
- 応用ソフトウェアごとにUIが異なっては、使い勝手が悪い
 - ファイルの保存ダイアログは、色々なアプリケーションで共通している
- 現在ではアイコンなどを利用したGUI (Graphical User Interface) が主流

OSの役割

1. 計算機の抽象化+サービスの提供

- OSがない状態よりも、プログラミングを「易しく」「機械非依存に」する

2. 保護, 安全性, 資源管理

- 複数のプログラム, 複数のユーザがいる環境で、計算機の「独占を防ぐ」、ユーザ間での「データの保護」

OSがいるおかげで出来ていることの例

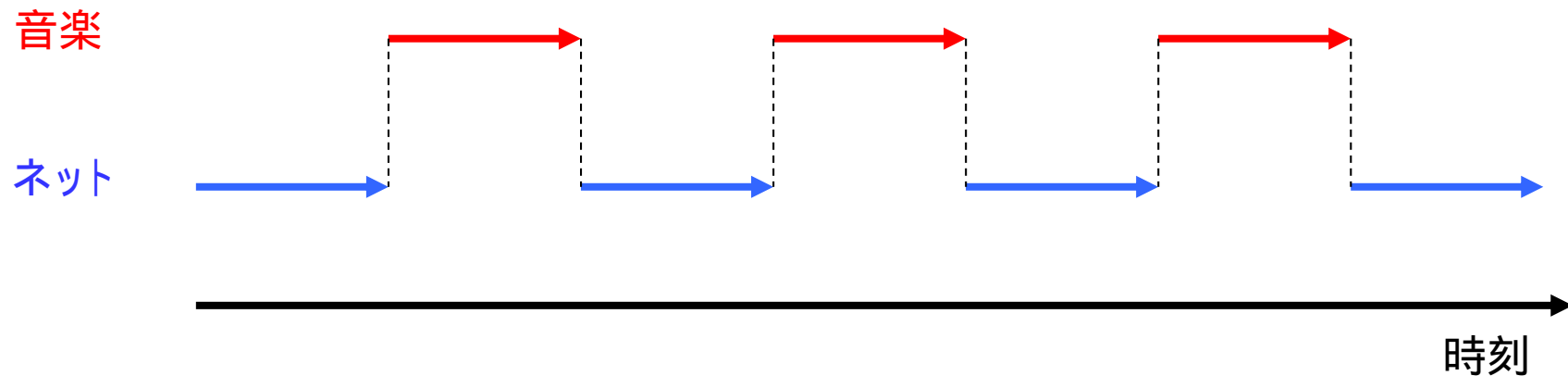
- 自分のプログラムはひたすら自分の計算だけをしているのに、複数・大量のプログラムが同時に起動・実行できる
- プログラムが暴走してもCtrl-Cで除去できる
- 他のプログラムとメモリ内に同居しているのに、メモリのどの部分を使えばいいのかわらなくてよい
- HDDだろうとCD-ROMだろうとUSBだろうと同じようにopen/read/writeで読み書きできる

計算機の抽象化+サービスの提供

- 「生のハードウェア(OSなしの計算機)」で、「音楽を聞きながらネットサーフィンができる」プログラムを書いたら?
 1. CPU : 複数のプログラムの同時(交互)実行
 2. メモリ : 物理メモリの割り当て
 3. I/O : プロセッサ外部(ディスク, ネットワーク)との通信

複数のプログラムの交互実行(1)

- (OSなしで)交互実行を保証するためにプログラムはどのように書かねばならないか?



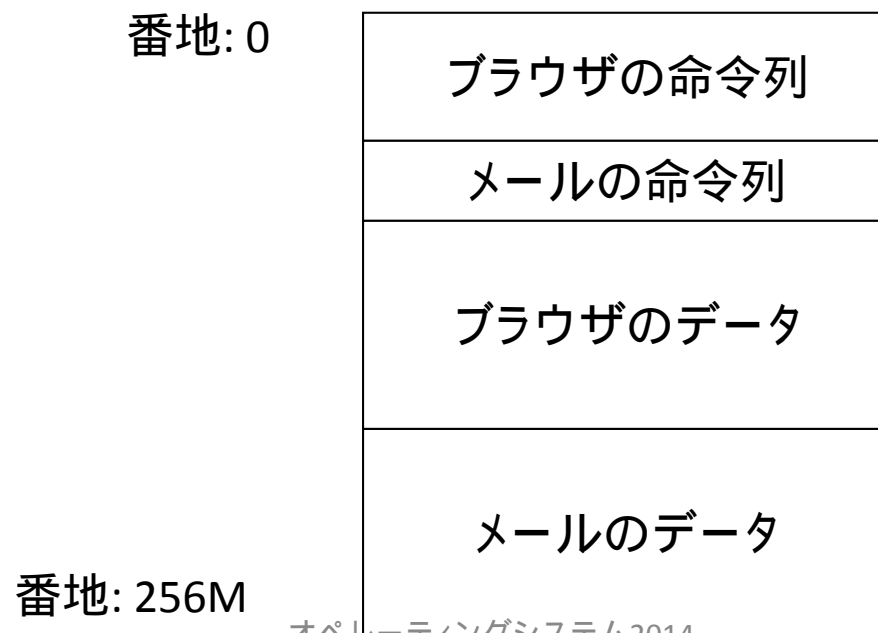
複数のプログラムの交互実行(2)

- こんなことをいたる所で...

```
draw_page() { /* ページを描画 */  
  for (...) {  
    do_some_drawing(); /* ちょっと描画 */  
    if ( $\Delta t > 100$  ms) { /* MP3Playerのために */  
      yield_CPU_to_MP3Player();  
    }  
  }  
}
```
- 実際には他にワープロ, ゲーム, etc.も走っているかも...
- OSの回答: スレッド, プロセス

メモリ管理(1)

- 有限(たとえば256MB)のメモリに, プログラム(命令列), データを重ねないように配置する



メモリ管理(2)

- ひとつしかプログラムがなくてもすでに複雑だが、その上複数プログラムがあると、
 - どのプログラムがどの部分を使うの？
 - その割り当ては立ち上げるたびに違うの？
 - メモリが足りなくなったらどうするの？
 - etc.
- OSの回答：仮想記憶，論理アドレス空間

複雑な入出力(1)

- 例：ネットワークからページをダウンロード
- `download_page()` {
 `send_request_to_server();`
 `receive_from_server();`
}

↕ > 100ms

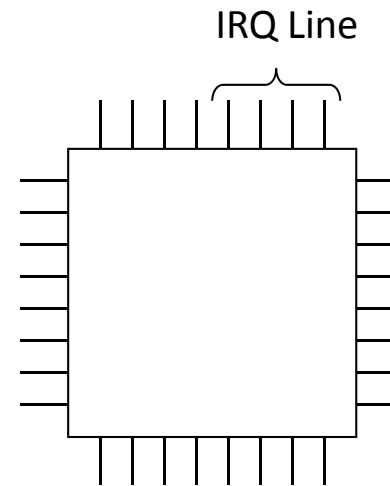


複雑な入出力(2)

- 入力待ちの間(より一般には「**することがない間**」)は,
 - 自分はCPUを使わない= 他のプログラムにCPUを譲る(yield)必要がある
- やがて入力 cameたら,
 - CPUはそれに「**気づく**」必要がある
 - ポーリング(定期的な検査)
 - ハードウェア割り込み(IRQ)

ハードウェア割り込み

- 外部装置からCPUへ送られる信号
 - CPUの制御(プログラムカウンタ)を強制的に、指定された番地(割り込みハンドラ)に移行
- 外部装置がイベントの通知に用いる
 - 入力データready
 - 出力可
 - 時間経過
 - ...

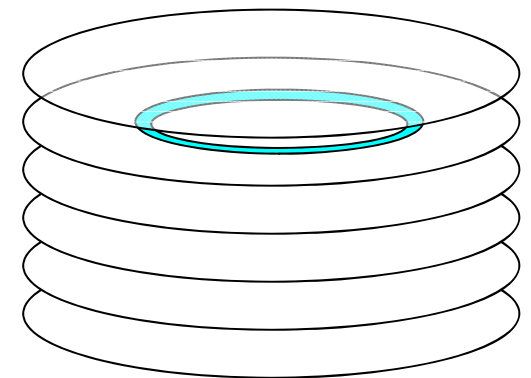


割り込みを用いた入力処理

- 以下のような割り込みハンドラ設定
 - `receive_from_network () {`
 - `write_data_to_browser_memory();`
 - `yield_to_browser();`
 - `}`

ディスク入出力

- ディスクの構造や容量に基づいたアドレス (e.g., (シリンダ番号, トラック番号, セクタ番号))を用いてアクセス
- メモリと類似の, 空き領域管理
- 必要な情報の格納位置を得るための情報
- ネットワーク同様の割り込み処理



「計算機の抽象化+サービスの提供」 まとめ

- CPU, メモリ, 入出力IF, 2次記憶 etc. が, 生のハードウェアそのままでは使いづらい
 - 複数のプログラムが同時進行しているときに顕著
- OSが「よりシンプルな」プログラミングインタフェース(API)を提供(“抽象化”)
- 「お役立ち道具集」としてのOS

保護, 安全性, 資源管理

- OSの「真の」存在理由: 資源を管理し, 実行中のプログラムを保護
 - 1つのプログラムがCPUを独占利用できない
 - 1つのプログラムが他の走行中のプログラムのメモリを破壊することはできない
 - あるプログラムが受け取るべきメッセージを他のプログラムが横取りすることはできない
 - 本来権利のない人が計算機をつかうことはできない
 - 本来権利のない人が他人のファイルを読み書きすることはできない
- 間違えても, または悪意があっても, できないようにする

第2回の講義内容

- ハードウェアとOS
- CPUとデバイス、割り込み、記憶装置、ハードディスク装置、RAID、パリティ
- スレッドとプロセス、メモリアドレス空間、ファイルシステム、NIC、ソケット、カーネル