

第三回 オペレーティングシステム

2002年9月19日

前回のレポート

- 問1の出題に不適切な部分 → 採点対象外とする
問1: 「-108」を1の補数と2の補数でそれぞれ2進法表記せよ。
有効桁は8桁とする。
- もとの出題意図は「108」の1の補数と2の補数
「+108」の補数は
 $108_{10} \rightarrow 64+32+8+4 \rightarrow 01101100_2$
1の補数 → 10010011
2の補数 → 10010100
「-108」の補数は負の数の内部表現形式を特定しないと答えられない
負の数が2の補数の場合: $-108_{10} = 10010100_2$
1の補数 → 01101011
2の補数 → 01101100 → 「+108」に戻る

単位と用語

- 計算機では小さな数から大きな数までを扱うので、単位の接頭語が欠かせない

k	10^3 キロ	m	10^{-3} ミリ
M	10^6 メガ	μ	10^{-6} マイクロ
G	10^9 ギガ	n	10^{-9} ナノ
T	10^{12} テラ	p	10^{-12} ピコ
P	10^{15} ペタ	f	10^{-15} フェムト
E	10^{18} エクサ		

単位の例外：「バイト」

- 二進法の一桁分を「ビット(bit)」とよぶ
- 8 bit = 1 byte
- $2^{10} = 1024$ を大文字の「K」であらわす
- 1 Kbyte (1KB) = 2^{10} byte = 1024 byte

1 KB \neq 1 kB (1000byte)

- 1 Mbyte (1MB) = 2^{10} KB = 2^{20} byte
= 1,048,576 byte
- 1 Gbyte (1GB) = 2^{10} MB = 2^{30} byte
= 1,073,741,824 byte
- 二進法的に「きりのいい数字」

通信速度の単位：「bps」

- 計算機と計算機を結ぶネットワークの通信速度の単位は伝統的に
ビット毎秒 (**bps**: bit per second)
が用いられる
- 計算機内部の磁気ディスク装置などのデータ転送速度は
バイト毎秒 (**B/s**, byte/s: byte per second)
が用いられる

1 Gbps = 125 MB/s

10 MB/s = 80 Mbps

オペレーティングシステムとは

- Operating System = 「OS」と略すこともある
- 計算機を動かすためのもっとも基本となるソフトウェア
- 「計算機をいかにして人間が扱いやすいものにするか」

応用プログラム
(アプリケーション)

コンパイラ、リンカ、ローダ、
エディタなど
サービスプログラム
処理プログラム

ジョブ管理
通信管理
ファイル管理
入出力管理

プロセス管理
記憶管理
ハードウェア管理
核 (kernel)

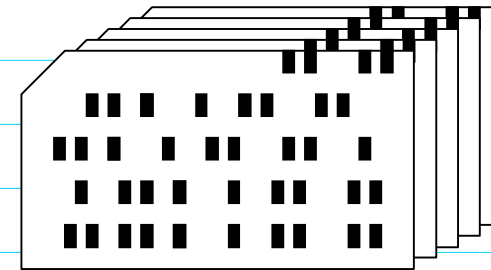
役割

- スループット(throughput)の向上
- 資源(resource)の管理
- 使いやすさの向上
- 信頼性などの向上
- ハードウェアの差異の吸収
- ソフトウェア間の連携
- 並行処理
- 情報通信への対応

オペレーティングシステム

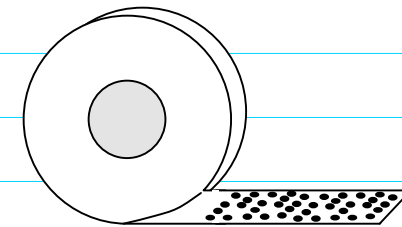
OSと利用形態の進展

- 大型計算機（メインフレーム）
 - バッチ処理・時分割処理
 - パンチカード、穿孔テープ
 - ラインプリンタ
 - インタラクティブ処理
 - CRT端末、スクリーンエディタ



パンチカード

- ミニコンピュータ・ワークステーション
 - UNIX・ネットワーク
 - X-window・マウス



穿孔テープ

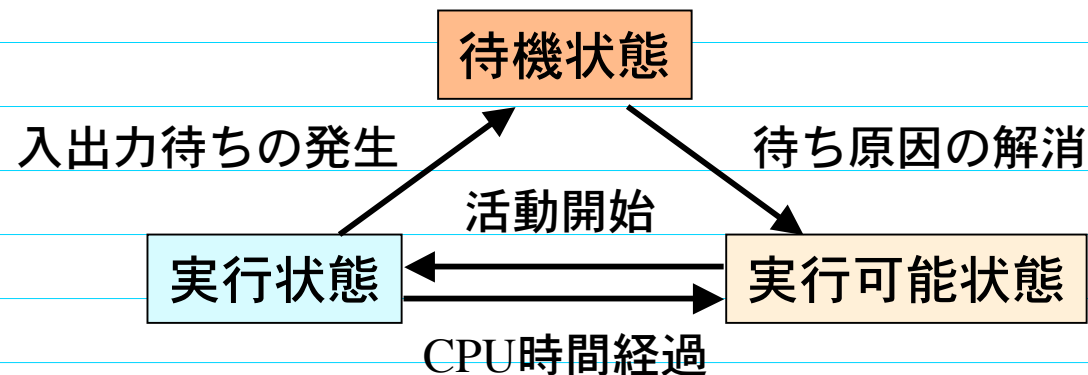
- パーソナルコンピュータ
 - DOS/V, Windows, Macintosh
 - シングルトask → マルチタスク
 - グラフィックユーザーインターフェース(GUI)・マウス
 - FreeBSD, Linux (パソコンで走る”UNIX”)

プロセス管理

- 一つの仕事(プロセス)を達成する時に磁気ディスクや人間のキーボード入力などの入出力などがあると、CPUはひたすら待ち続ける
(CPUのクロックが 1GHz → 1秒間に10億回の計算能力)
- 遊んでいる時間を他の処理に回すことで、同時に複数の処理を並行して実行させることができる
(Windowsのタスクマネージャー → 使っていないときでも複数のタスクが走っている)
- 「ワープロ文書を印刷しながら電子メールを送信し、ホームページを見る」
- 一台のCPUが複数のプロセスを切り替えながら処理している
→ マルチプロセッシング、マルチタスキング

プロセスの状態

- それぞれのプロセスは「実行可能状態」「実行状態」「待機状態」のいずれかの状態
- 実行状態にあるプロセスは一つだけ
- 入出力でなんらかのイベント待ち → 待機状態に
- 次の実行可能状態のプロセスの一つが実行状態に
- 待機中のイベントが満たされると → 実行可能状態に
- 一定時間以上CPUを使用しても実行可能状態に戻る



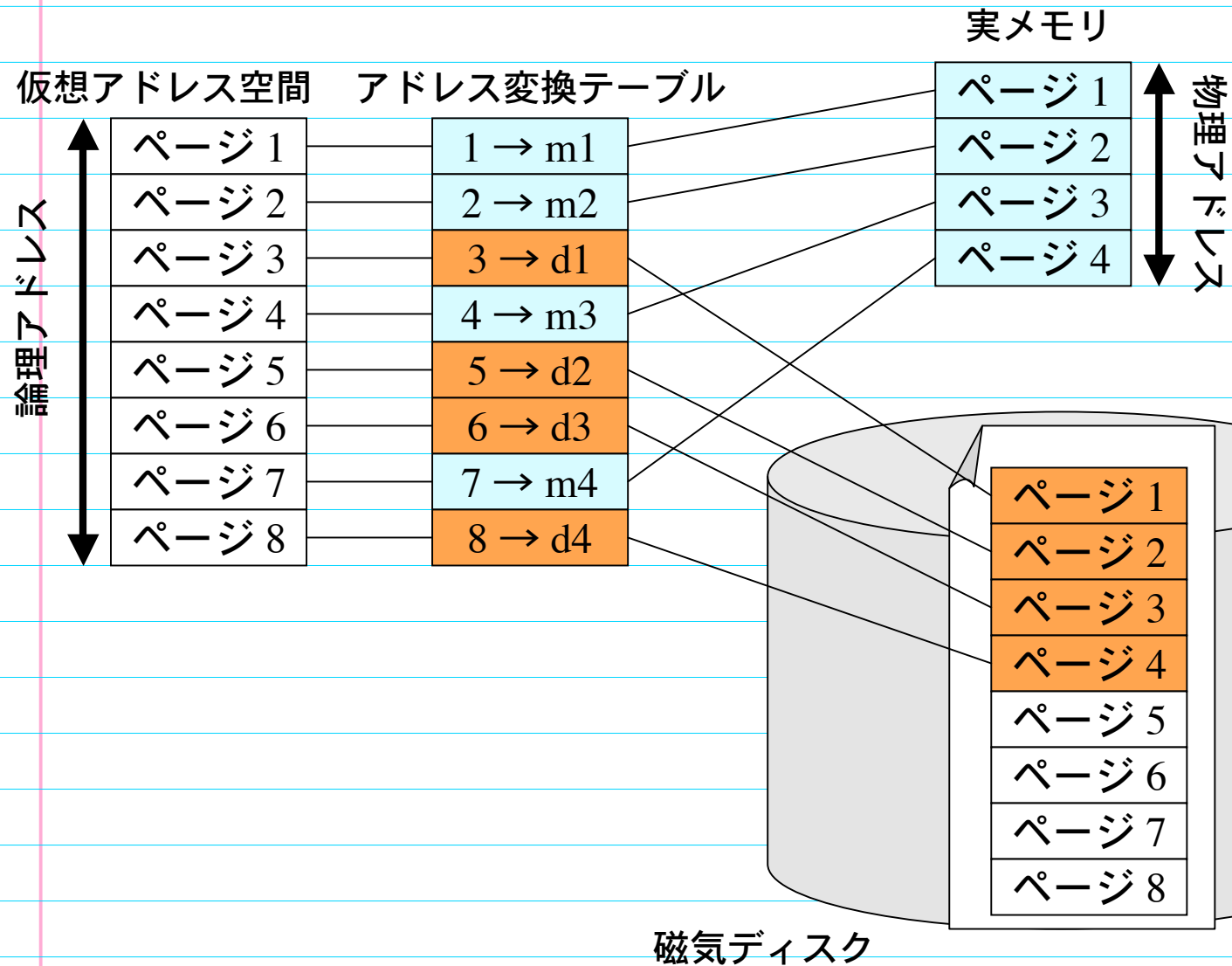
メモリ管理

- OSは常に現在のメモリ(主記憶)の使用状況を把握
- 必要に応じてメモリ領域の割り当て・解放
- 各プロセスがお互いのメモリ領域に干渉しないように、許可された領域以外の記憶を保護
- 半導体メモリは磁気ディスクよりもアクセス速度が速いが単位値段あたりの容量が少ない
- 主記憶で不足するメモリ領域を磁気ディスクで補う
仮想記憶方式が一般的
- 大きなメモリ領域を必要とするプログラムを経済的に実行できる

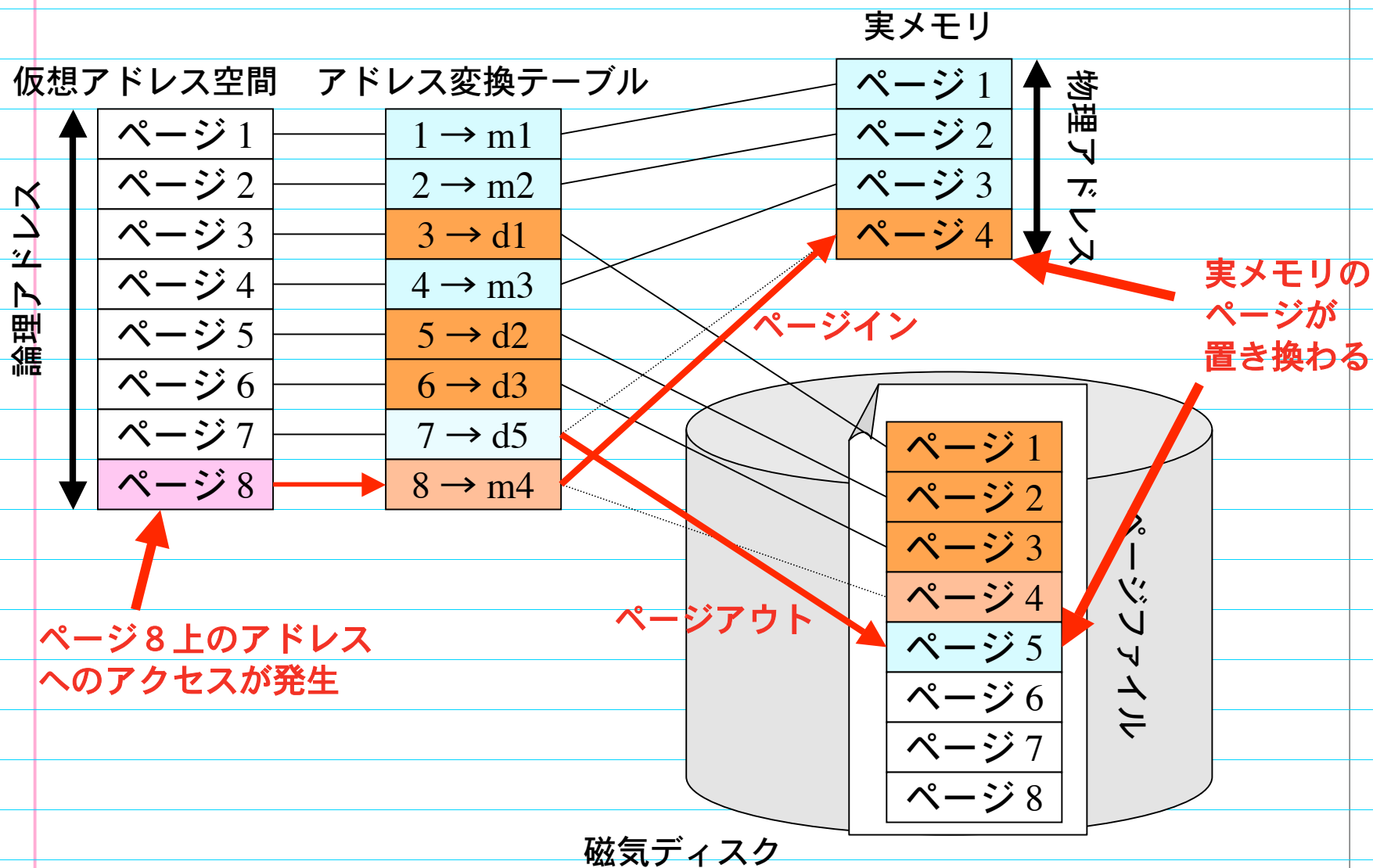
仮想記憶方式

- CPUが利用するアドレス(論理アドレス)と実際のメモリアドレス(物理アドレス)を区別
- 一定の大きさのページ(512B~ 4KB) に分割して管理
- 頻繁にアクセスされるページは物理メモリ上に、それ以外は磁気ディスク上のページファイル(スワップファイル)に置かれる
- ページファイル上のアドレスにアクセスする時は物理メモリ上のページと入れ代わる (ページングまたはスワッピング)
- ページングが頻発するとパフォーマンスが極端に落ちる → 物理メモリを増設するとよい (Photoshopなどの画像処理ソフトやマルチメディアのデータを扱うアプリケーションなど)

仮想記憶方式

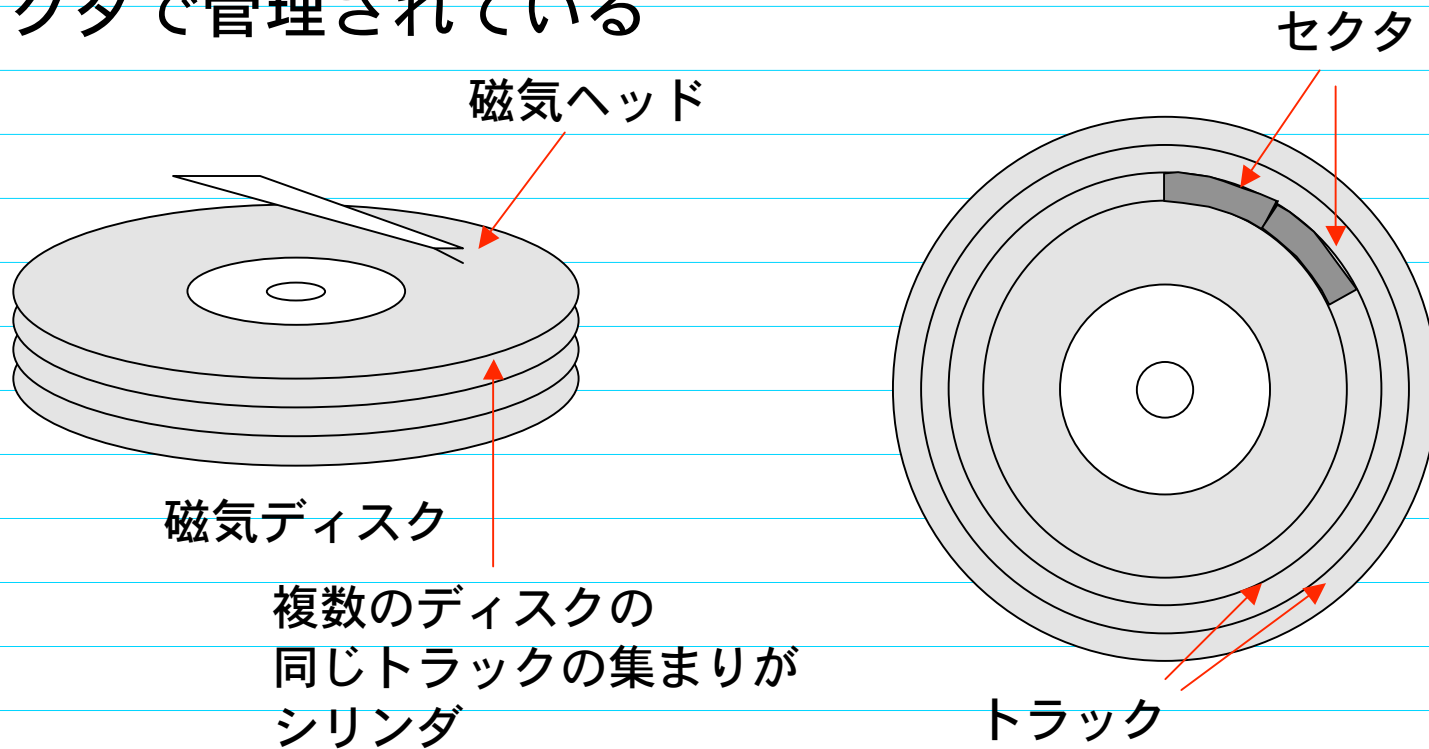


ページング(スワッピング)



ファイル管理

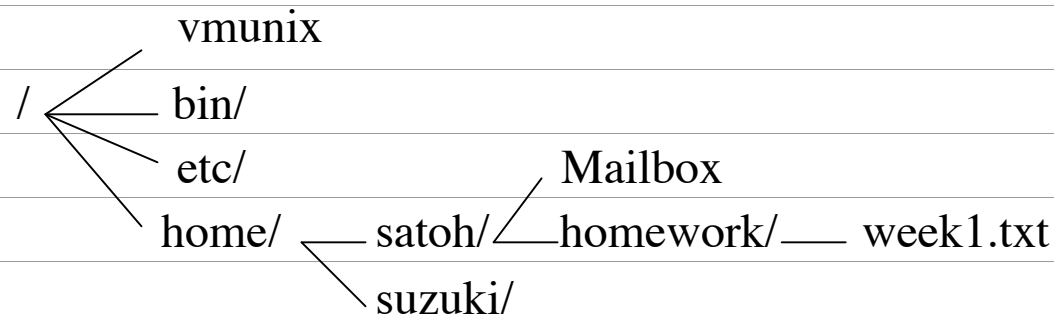
- 磁気ディスク上のデータはシリンダ・トラック・セクタで管理されている



- プログラムやデータごとにひとまとまりの情報として補助記憶装置から主記憶にロードできる単位をファイルと呼ぶ

ファイル管理

- ファイルには「名前」「種類」「大きさ」「作成日時」などの情報があり、ディレクトリ(directory)という登録簿に登録されている
- ディレクトリはサブディレクトリを含む階層構造にすることができる
- OSは補助記憶上のファイルを管理したり、探索や更新などの操作機能を実現する



UNIXファイルシステムの例

ファイルとrecord

- Record : ファイルに記録する際の情報の単位
例: 住所録で一人ごとの住所、電話番号などの記録
プログラムの一行ごと
- 可変長recordのファイル
→ 最初のrecordから順次読み書きするだけ
逐次ファイル (sequential file)
- 固定長recordのファイル
→ 任意の順序でrecordにアクセスできる
ランダムファイル (random file)

ユーザーインターフェース

- 人間が計算機を操作するための入出力の管理機能
- CUI : Character User Interface 「クーイ」
 - コンソールや kterm などの端末ソフトからキーボード操作
 - ls, pwd, vi, mule などの文字列を「コマンド」として入力
 - コマンドやオプションを覚えるのが難しいが、一度覚えたら複雑な操作を組み合わせて自由に扱うことができる (スクリプト)
 - メインフレーム、UNIX, MS-DOSなど
- GUI : Graphical User Interface 「グーイ」
 - マウスやタッチ패드などのポインティングデバイス
 - ウィンドウやメニュー、アイコン、デスクトップ
 - 深く考えずに直感的に操作することができる
 - コピー・ペーストなどの操作感の統一
 - UNIX/X11, MacOS, Windowsなど

OSとブートストラップ

- プログラムをファイルから読み込んで実行させるのはOSの役割
- ではOSを起動するのは誰?
- 計算機のハードウェアには、電源投入後にOSの先頭部分を磁気ディスクから読み込んでスタートする機能がある(PC DOS機 → MBR: Master Boot Record)
- 先頭部分は引き続きOSの後の部分を読み込んで順次実行していく
- 靴ひもをつかんで自分で自分を引っ張りあげる様子になぞらえて「ブートストラップ」と呼ばれる
- 「システムをブートする」「システムを立ち上げる」「システムを起動する」ともいう

OSの種類

- 世の中に広く普及したOSの種類はあまり多くない
「みんなが使うOSほど便利になっていく」→ 淘汰
- Linux以外はほとんど米国を中心に開発された
- かつては計算機メーカーのノウハウの結晶
→ 市場の覇権 cf. 1982年「IBM産業スパイ事件」
- メインフレーム: IBMのOSが事実上の標準
- ワークステーション: 様々なUNIXの亜種
- パーソナルコンピュータ:
 - CP/M - Basic環境 → MS-DOS → Windows (Intel系)
 - Apple, Apple II → Lisa → MacOS (Motorola系)
 - FreeBSD, Linux (Intel/Motorola/Alpha/その他)

OSの種類（続き）

- 日本では東京大学の坂村教授が提唱する「TRON」が組み込み機器などのOSとして普及：仕様が公開されている

<http://www.tron.org/>

- Linuxはwebやネットワークの普及が生み出した「オープンソース」なOS → 世界中から様々な技術者や有志が参加して日夜改良が続いている → プログラムが読める人なら誰でも中身を知ることができる

<http://www.linux.org/>

あらゆる種類のハードウェアに移植されつつある

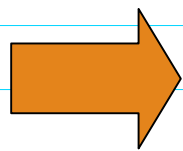
- 企業秘密の開発ベースからオープンな開発ベースへ「オペレーティングシステムの”民主化”の時代」？
- Linux以前は NetNews などによる情報交換がUNIX発展のベースとなっていた

UNIXの歴史

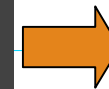
- 1969年に米国AT&T社のベル研究所でKen Thompson, Dennis Ritchieを中心に「趣味的な興味をきっかけに」作られた時分割多重処理OS
- 当初、教育機関向けのソースコードライセンスが安価で販売され、研究開発が進んだ
- 1980年前後からAT&T版(System V, SysV)とバークレー版 (BSD 4.x)の二つの潮流に分かれる
- それ以前のOSに比べると異なるハードウェアへの移植性が高いために、研究開発が途切れることなく進展した
- 世界中で開発されたプログラムがカリフォルニア大学バークレー校にフィードバックされ進化していった

UNIXの思想

- スモール・イズ・ビューティフル
 - 一つの仕事を上手にこなす小さなプログラムを作る
 - 小さなプログラムを柔軟に組み合わせて複雑なことができる
- 10%の努力で問題の90%を解決する
 - 複雑な問題はあえて無視することで簡潔で汎用なシステムを作る
- 迷ったらいちばん単純な方法を選ぶ
 - 単純なツールほど信頼性もよく、理解も楽、使うのも楽
- マシンに頼らず問題を解決する
 - 土台になるハードウェアをできるだけ無視することで移植性がよくなる
- 適正なレベルで解決すれば二度とやらないですむ
 - 一か所で解決された問題をどこでも利用する



移植性がよく、時代を超えて
研究者に愛されるOSとなった



Linux文化
の原形

今週のレポート

- 問1: 十進法の「75」を1の補数と2の補数でそれぞれ2進法表記せよ。有効桁は8桁とする。
- 問2: 「ADSL 12M」などの広告で見られる通信速度はバイト毎秒で表すといくつになるか。
- 問3: テキストファイルの改行に用いられるコードはOSによって異なる。Unix, Windows(MS-DOS), MacOSで一般的に使われる改行コードはそれぞれ何か。16進法で「0x2a」のように表記せよ。
- 問4: マウスを発明したのは誰か。また、マウスの発明によって計算機の使い方はどのように変わったか。検索して分かったことを記せ。
- 問5: Linuxの開発の歴史について検索して分かったことを記せ。参考にしたwebページのURLも記せ。

参考文献

- 稲垣耕作 著「コンピュータ科学の基礎」コロナ社
ISBN4-339-02338-8
- 小舘香椎子, 上川井良太郎, 中村克彦 共著
「教養のコンピュータサイエンス情報科学入門 第2版」丸善
ISBN4-621-04871-6
- 菊沢正裕, 山川修, 田中武之共著「情報リテラシー：メディアを
手中におさめる基礎能力」森北出版
ISBN4-621-04871-6
- Don Libes, Sandy Ressler共著 坂本文 監訳 福崎俊博訳
「Life with UNIX」アスキー出版
ISBN4-7561-0783-4