

# オペレーティングシステム

第6回(2009.05.21)

メモリ管理

# アパート(マンション)への入居

- アパートなどを借りる
  - バスなし、トイレ共同(今はほとんどない)
  - バス・トイレつき(今は当たり前になった?)
    - ワンルーム
  - 2DK、2LDK、3LDK...
  - ベランダ付
  - (共有の)ロビー・プール付
- ホテルに住む?
  - ホテルの部屋を占有する?
- 居住する「空間」は、誰のもの？

# 共有空間と占有空間

- 全体の効率として、「共有」でも十分なもの
  - 高級マンション？での、ロビーやプール
    - 昔は、安アパートでは「トイレ」やシャワールームも「共有」だった
- どんなに狭くても、占有したい空間
  - ベッドルーム（寝室）
    - インターネットカフェ？の個室？
      - 占有か？共有か？

# プログラムとメモリ

- プログラムの実行中は、メモリ上に呼び出される。
  - 「占有空間が必要」
- 実行が終了すれば、「解放」されて、他のプログラムに使用される。
  - 実行中でも、WAITING状態になると、「追い出される」ことがある。
    - P90 プロセスの実行状態
      - 実行中 / 「待ち」状態 / “ready”状態
- BIOSなどは、「持ち家状態」で、その場所からは動かない！

# メモリ管理

- 教科書P117
- CPUの命令は、「主記憶」から「データ」を読み出し、主記憶にデータを書きこむ。
- 「プログラム」も「主記憶」から読み出して実行する。
  
- 「主記憶」の領域が割り振られて、初めて実行することができる。
- 「主記憶」にメモリを割り振られていない状態
  - = 住民票を届け出していない状態
    - 選挙権も、学校に通うこともできない……

# メモリの分割

- プログラムが必要とするメモリ領域
  - プログラムのロード領域
  - 「プログラム中」で要求されることがはっきりしている、初期データ領域(変数名・配列など)
  - 動的にプログラムから割付要求が出る可能性がある領域:(Heap)
  - スタック領域
  - 予備領域
  - 教科書P118

# メモリの静的割付と動的割付

コンパイル時にメモリ領域が確保される。

プログラム＋初期データ領域

⇒ 静的割付領域 (static allocation)

動的割付 (dynamic allocation)

P123

ヒープ領域(Heap領域)と呼ばれる。

malloc(), calloc(), realloc()などの関数

メモリを割り付けて、領域を変数に割り振る。

free()

割り付けられたメモリを解放する。

# メモリ割付アルゴリズム

- 教科書P124
  - ビットマップ方式
    - 空き領域の地図がある
  - リスト方式
    - 空き領域のリストを作成する。
- 検索アルゴリズム
  - First Fit
  - Best Fit
  - Worst Fit



# メモリの断片化

- Memory Fragmentation
- どんなに、アルゴリズムが優秀でも、割付・解放を繰り返していくと、中途半端に残った領域が出てくる。
- 断片化したメモリは、「詰め直し」を行う場合もある。

# メモリリーク

- Memory Leak
  - メモリが漏れる？
  - プログラムのバグで、「割付」したメモリを解放し忘れたり、動的に割り当てられたメモリが解放されなかったりした場合に、この分が、どんどんと減っていく現象
- メーンフレーム(大型コンピュータ)で経験を積んだプログラマは、マナーが悪い！
  - ファイルは開きっぱなしでも、OSが閉じてくれる。
  - メモリは割り付けっぱなしでも、OSが解放してくれる。
- ところが、Windowsは閉じて解放できなかった。
  - XPから改良された。
    - OSが責任を持って解放する。

# OSのメモリ不足？

- オペレーティングシステムが、「メモリ割付エラー」を起こす！？
- 合計としては「空き容量」は十分だが、全体が小口に分かれてしまって、割付できない場合：「メモリ不足の」エラーが帰ってくる。
  - OSのアルゴリズムの問題

# ガーベージコレクション

- 「ごみ拾い」P123
- 似たような処理に、ハードディスク上のファイルを集める「defrag」がある。
  - フラグメンテーションを解消する
- 断片化したメモリをかき集めて、「まとまった領域」を残すように、再配置を行う。

# オーバーレイ

- 一度にメモリに入りきらないプログラムを実行する手法
- 「仮想メモリ」が使われるようになってから、あまりこの手法は使われなくなった。(アプリケーションプログラムレベル)
- 教科書P128

# スワッピング

- 主記憶には「現在実行中」のプログラムのみ配置するようにする。
  - 「待ち状態」に入ったプログラムやデータは、直ちに「ファイル」に書き込まれて、主記憶から消される。
  - Swap: 交換すること
  - 教科書P129
- 「仮想メモリ」でも頻繁に行われるようになった
- 元々は、「端末用のメモリ管理」

# 仮想メモリと物理メモリ

- 「実装しているメモリ」・・・ 物理メモリ
- プログラムに「仮想メモリ」を見せて、実際の物理メモリよりも大きいプログラムや、データを扱えるようにしているもの。
  - 「仮想メモリ方式」： P131
- ハードウェア自身が、この仕組みを持っている。
  - Intel: 80x86CPUのシリーズ：
    - アドレスを表現するのに、セグメント+オフセットの概念を使用した。
    - セグメントアドレス ⇒ メモリのページ化に貢献
  - Motorola: 68000シリーズのCPU
    - セグメンテーションを行わなかった。
    - アセンブラレベルのプログラムは、非常に容易だった・・・。

# アドレス空間とアドレスの割付

- アドレス空間は、アドレスバスの容量いっぱいまで確保できる。
  - 実際の記憶素子が実装されているとは限らない。
  - 「メモリ」としてソケットに差し込むと、ハードウェアが認識して、「何番地」の「実アドレス」にメモリが存在しているか把握される。
- プログラムは、「実際に装着されている記憶素子の容量」を一切意識することなくプログラムできる。
- OSが、アドレスを変換して、「実アドレス」にマッピングする。



# ページング

- Pageには、「呼び出す」という意味がある。
  - 日本で言う「ポケベル」は、英語ではPager
  - 本のPageは、読みたい内容がどこに書かれているかを、すぐに「呼び出す」ためのもの
- 日本でページ(頁)と言うと、一枚の紙を指すことがある。
- ここで言うPageには、両方の意味が含まれている、と考えるとわかりやすい。
- メモリを「ページ」に区切って管理し、呼び出す。
  - Page呼び出しは、“Paging page”になる・・・!?

# アドレス変換のアルゴリズム

- 教科書P138～P141
- アルゴリズムの最適化には、常に「検討の余地」がある。
- 「何のためにアドレスを変換するか」
  - 実アドレスが存在しない「仮想アドレス」があるから。
  - 「アドレス変換エラー割り込み」
    - ⇒ 「エラー」にせず、「新規割り振り」を行う。
- 永遠の研究課題
  - ハードウェアも改善の余地が、常にある。

# メモリスケジューリングのアルゴリズム

- 教科書P143～P147
- どのように「メモリ」を割り当てるか!?
- 常に、トレードオフがある。
  - トレードオフって？
    - 「あっちを立てれば、こっちが立たず」
    - 「長所」と「欠点」が裏表になっているような状況
    - 「ダイエット中に、彼女(彼氏)に「食べ放題」に誘われた。どうしよう。」
- 「快適さ」(どんなプログラムも、サクサク動く)の追求は、難しい。(犠牲を最小限にできるか、が課題となる。)

# スラッシング

- Thrashing

- 仮想記憶環境下において物理メモリが不足しオペレーティングシステムがディスクとの入出力に処理能力の殆どを奪われてしまった状態を指す。(Wikipedia)
- 多数のプログラムを起動したり、「重い」プログラムを走らせると、ハードディスクのアクセスランプが点滅しっぱなしになり
  - マウスの反応は鈍くなって、画面上で移動しなくなり、
    - キーボードを押しても反応しなくなり、
      - 電源ボタンを押しても反応しなくなり、
      - もう何をやっても、どうしていいやら
- ⇒で、ファイルが壊れるのを覚悟しながら、電源ボタンの「3秒押し」しか、方法がなくなること

# まとめ

- OSは、メモリ(「主記憶」)を複数のプログラム(プロセス)に割り付ける。メモリを割り付けられないと、プログラムは実行できない。
- 動的なメモリの割付では、以下のような問題が起きる。
  - フラグメンテーション(断片化)
  - メモリリーク
  - スラッシング
- 古くは、メモリの容量の問題を解決するために、以下のような技術を使った。
  - オーバレイ
  - スワッピング
- 仮想メモリシステムが、最新の解である。
  - ページングや、メモリスケジューリングのアルゴリズムの開発が、課題となる。