

## コンピュータ概論B

－ ソフトウェアを中心に －

### #03 プログラムの実行形態

京都産業大学  
安田豊

## OSの管理機能

- 教科書 pp.82
- 主要な仕事 = 管理
  - － プロセス管理 = プログラムをどう実行するか
  - － メモリ管理 = メモリをどう利用するか
  - － ファイル管理 = ファイルをどう扱うか
- イメージ湧きますか？
  - － そもそもプログラムの実行形態がわからないとわからないことばかり

## プログラムの実行

- ジョブ(job)とタスク(task)
- バッチ処理 (教科書 pp.87)
  - － 最もシンプルなプログラムの実行形態
  - － 電源を入れる
  - － OS をメモリの中に読み込ませる
  - － ( )あるプログラムを読み込ませる
  - － そのプログラムの先頭から実行させる
  - － 終わったら から繰り返し
  - － 最後は電源を切る

## バッチ処理

- 特徴
  - － メモリ中には実行中のプログラムしかない
    - メモリをどう使うか考える必要がない
  - － 順次 (逐次) 実行
    - 全力で今のプログラムを実行
    - 他の処理をどう実行するか考える必要がない
    - 次のプログラムはその実行が終わるのを待つしかない

## バッチ処理

- 欠点
  - － 効率を上げるには実行順序に配慮が必要
    - 小さなプログラムは先に掛けるなど
  - － 自然に結果もひとまとめで受け取るように
    - BATCH: ひとたば、一括 (ひとくくり)
    - 一括結果を後で受取るタイプの処理に向く
- 美点
  - － 全力処理
    - 定型処理なら何時間後に終了するかがわかる

## バッチ処理

- 典型的な利用形態
  - － カード入力、プリンタ出力
  - － オペレータ付き
  - － 1970年代からの大型データ処理機
    - 大型汎用機、メインフレームと呼ばれるタイプ
  - － 伝票処理など
    - 銀行システム (夜間に伝票を処理)

## バッチ処理

- ゲーム
  - 内部的には処理形態は似ている
    - 単一のプログラムを全力で実行
    - が、バッチ処理、バッチ的とは言わない
  - 理由：
    - 結果をひとまとめで貰うのではなく
    - 対話的な操作が目的（操作自体が処理結果）
  - リアルタイムシステムに近い（後述）

## バッチ処理の苦悩

- マルチタスクでありたい
  - 短い仕事を割り込ませたい
  - 長い仕事を待つだけなのはイヤだ
  - 人間はそうやって処理をしている
  - タスクのマネージメントをやりたい
- Task
  - コンピュータにとっての仕事の単位
  - バッチ処理はシングルタスク
  - マルチタスクでありたい

## タイムシェアリング処理

- マルチタスクの実現
  - 複数のプログラムをたらい回しに処理
  - 教科書 pp.87 の図参照
- 複数ユーザでの同時利用
  - 多くの端末機を接続
    - 複数人が「同時に使っている」感覚に
    - 「キーを打つ」「表示する」という「短い処理」を割り込ませている
  - バッチスタイル
    - 一台のコンソール+一人のオペレータ

## タイムシェアリング処理

- TSS / Time Sharing System / 時分割システム
  - CPU 処理の時間を細かく分割
  - 特にCPUが空いているときにチャンス
  - コンピュータの全ての部品が全ての時間全力で機能しているわけではない
    - 計算をしている間はディスクが暇
    - ディスクが回っている間はCPUが暇
  - マルチタスクはコンピュータの処理効率を上げる（可能性がある）

## タイムシェアリング処理

- まとめ
  - マルチタスクの一つの実現手法
  - 一台のホストに端末を多数つなぐというスタイル
  - コンピュータの処理効率を上げられる可能性
- 注意
  - 「あたかも一人で占有し」というフレーズ
    - 教科書 pp.88
    - バッチ処理に対してそう見えた、というだけ
  - 「複数人で共有する」が妥当な表現
  - その実体は「忙しくたらい回し」という事

## トランザクション処理

- 教科書 pp.88
- 一連の処理
  - 座席予約、銀行振り込みなど
  - 取引の発生とともに即時実行する必要のある小さな処理
  - トランザクションと呼ぶ
  - オンライン処理とセットで発展

## オンラインシステム

- 通信回線で結ばれた端末処理
  - 銀行のATM/CDや窓口端末など
  - トランザクション処理が目的
- みどりの窓口 1960 国鉄+日立
  - <http://www.kahaku.go.jp/special/past/jyoho/ipix/3/3/06.html>
  - 世界最初期のオンラインシステム

## オンラインとトランザクションの甘い関係

- みどりの窓口
  - キップを売るということは、
  - 空席確認、同時に確保
    - この処理を即時に実行する必要がある
  - このような処理をトランザクションと呼ぶ
  - オンライン無きトランザクション処理とは？
    - ちけっとびあ？（電話+人間）

## ホスト・端末 オンライン・スタンドアロン

- 集中ホスト+多数端末という図式
  - 座席情報の管理は集中ホストでやるしかない
  - ホスト+オンライン端末+トランザクション
  - 当然マルチタスク（TSSで実現する場合もある）
- スタンドアロン (Stand Alone)
  - オンラインに対する用語
  - 一台だけで処理を完結させているコンピュータ
  - パソコンなど（大型機でもあり得る）
  - 最近ではネットワークから切り離されている場合のことをそう呼ぶ事も多い

## 分散処理

- 教科書 pp.89
  - 多くのコンピュータを配置（自然に離れる）
  - ネットワークで相互接続
  - 連携して一つの処理を完結
- なぜ分散処理か
  - ネットワークの高速化
  - 単体コンピュータの低廉、小型高速化
  - 地球シミュレータなどスパコン用途などにも展開
- OSとの関係
  - 従来のOS：単体コンピュータ内の処理を支援
  - 分散OSもある

## リアルタイム処理

- 教科書 pp.88
- 特徴
  - Real Time：実時間処理
  - マルチタスクも可能
  - イベント発生による割り込み方式
  - 一定時間以内（次の歯車が回る前に）に必ずその処理が終わる

## リアルタイム処理

- 適用分野
  - 複数組み合わせられた機械の制御など
  - ゲームもこれに近い
    - 次の画面まで(1/30sec)に一連の処理を完了
- リアルタイムOS
  - TSSで実現すると処理終了時間が不測
  - 処理完了時刻が一定以内の誤差で確定できるOSが必要
  - TRONなど工業用、通信用に多数

## TSS vs RTOS

- Windows / Mac / Unix (Linux) は
  - すべてマルチタスクOSである
    - そしてTSSである
  - ユーザがマウスをクリックしたら、、、
    - CPUの処理を細かく区切って少しずつ処理
    - 多くの処理を行っているため反応が遅くなる
    - 優先度設定などでごまかしている
- (いわゆる) デスクトップOSには？
  - RealTimeの方が良いのではないか？
  - 今後動画再生などRTOS向きのものが増える
  - TRONベースのOSもある