

# メモリ管理の基礎知識

～ここはだれ？私はどこ？とならないために～

## とっちゃん

高萩 俊行

Microsoft MVP for Windows-SDK 2005/10-2007/09



わんくま同盟 大阪勉強会 #12

## アジェンダ

- はじめに
- アロケータとは
- アクセス方法
- アロケータの影響範囲
- まとめ

はじめに

- メモリには...
  - 変更できる...RAM
  - 変更できない...ROM
- メモリに対する操作には
  - メモリを読む
  - メモリに書く
  - メモリで実行

## はじめに

- プログラムで利用できるメモリ
  - データエリア
    - ヒープ
    - スタック
  - 初期化済み変数エリア(特殊なデータエリア)
    - 初期値を持つ初期化済み変数
    - 初期値を持たない初期化済み変数
  - コードエリア(実行権限を持つデータエリア)
    - DEP(Data Execute Prevention:データ実行防止)
      - データエリアとコードエリアは違うものである

## アロケータとは

- 動詞の Allocate を英和辞書で...

- [III[名]([副])] ...を(特定の目的のために)とっておく, 計上する((for ...));

[allocate B A/allocate A to B]〈A(資金・土地・責任・費用など)をB(人など)に〉割り当てる, 配分する, あてがう

引用元:プログレッシブ英和中辞典(Yahoo)

- コンピュータ的には...

- A=メモリ...かな?

- B=プログラム...かな?

- では Allocator or Allocater は？

- ありません！ Allocation が名詞形

- とあるライブラリには、Allocator という名のクラスはある

## アロケーターとは

- アロケーターとは...
  - メモリを割り当てるもの
    - 割り当てるとは？
      - 管理する
  - 管理する＝責任を持つ
    - どこに対して責任を持つ？
    - だれに対して責任を持つ？

## アロケータとは

- ハードウェアレベルのアロケータ
  - OSに対して
  - 物理メモリにたいして
- OSレベルのアロケータ
  - プロセスに対して
  - プロセスに割り当て可能な論理メモリ
- プログラムレベルのアロケータ
  - 個々のデータに対して
  - ヒープメモリ

## アクセス方法

- スタックメモリを利用するデータとは...
  - 寿命の限定された一時的な記憶領域
  - 寿命はメソッドから抜け出る時点まで
  - 通常は自動変数用の領域とプログラムの実行に必要な情報を格納
- ヒープメモリを利用するデータとは...
  - 実行時に必要に応じて利用するデータ
  - メソッドから抜けても、寿命を終えるわけではない
  - 通常は何らかの参照型変数として表現される



## アクセス方法

- 参照型というけれど...
  - 参照型ってなんなのさ？
    - 実体を指し示す型の総称
    - ついになる用語に値型がある
  - CやC++ のポインタとは違うの？
    - ポインタは参照型の一形態
- メモリアドレスとは違うの？
  - 同じに見えても違うもの！
  - アドレスは位置なので型情報を持たない！

## アクセス方法

- アロケータから受け取る情報
  - 受け取るための器が必要
    - 受け取るものは？
      - 参照型の変数
    - いらなくなったら？
      - どこかで不要を明示する

## ちょっと脱線

- 型とは...

- 型には2つの意味がある

1. オブジェクトの形状を表現する型
  2. オブジェクトの状態を表現する型
- 今日やったのはどっち？

- データの受け渡し

- 値渡し＝値を引数に入れる
- 参照渡し＝オブジェクトの**位置情報**を引数に入れる

## 管理範囲

- アロケータオブジェクトは一つではない
  - ランタイムライブラリごとに一つ！
    - ダイナミックリンク
    - スタティックリンク←.NET Framework にはない！
- アロケータの影響範囲
  - 基本は？
    - プロセス単位？
    - モジュール単位？
      - ランタイムライブラリって言ったような...

## 管理範囲

- MFCだとどうなるんだろう？
  - MFCのヒープ管理は非常に複雑に感じるが...  
基本は唯一つ！

# CWinThread の スコープ

それだけです！

## まとめ

- **メモリの種類**
  - あらかじめ用意されたメモリ
  - 実行時に必要に応じて用意するメモリ
- **アロケータとは**
  - メモリを管理するオブジェクト
- **影響範囲**
  - ランタイムライブラリの影響範囲

## まとめ

- なぜ「ここはだれ？」
  - ここ(この場所)はどこなのかでよいのか？
  - だれが用意したのかではないか？
- なぜ「私はどこ？」
  - 指し示す先はなにかでよいのか？
  - どこを指しているのかではないか？

## まとめ

- メモリは借り物である
  - 常にだれから借りているかを意識
- すべては寿命を持つ
  - 寿命を把握するのがメモリ管理の基本
- 寿命は管理するのではなく、把握する
  - 状況を把握することが大事