

オブジェクト指向分散環境OZ++の実行モデル

新部 裕 (三菱総合研究所*) 田代 秀一 (電子技術総合研究所) 鈴木 敬行 (シャープビジネスコンピュータ*)

5B-8

平川 秀忠 (日本ユニシス*) 塚本 享治 (電子技術総合研究所)

* 情報処理振興事業協会「開放型基盤ソフトウェア研究開発評価事業」研究員

1 はじめに

オブジェクト指向分散環境 OZ++では、ネットワーク上に分散したオブジェクトにより計算が行なわれる。

OZ++では、電総研で開発された分散システム OZ+[1]を基本として、多重継承のクラスシステム、およびマルチスレッドによるメソッドの並列実行の機能を追加している。

ここでは、OZ++のオブジェクトの実行モデルについて、マルチスレッドによる並列実行、排他制御/同期、メソッドの動的な束縛、オブジェクトのコピーイン/アウト、およびプロセスを説明する。あわせて、その言語上での表現についても触れる。

2 オブジェクトモデル

OZ++ではオブジェクトをグローバル/ローカルの二種類に区別して取り扱う。外部から利用され、共有されるオブジェクトがグローバルオブジェクトであり、グローバルオブジェクトの部品として使われるオブジェクトがローカルオブジェクトである。

言語では、変数の属性としてグローバル/ローカルを区別し、異なる型として扱う。グローバルオブジェクトの生成はグローバル属性を持つ変数に new することで行ない、ローカルオブジェクトの生成はローカル属性を持つ変数に new することで行なう。

```
global Object g; // グローバル
Object l; // ローカル
g->new (); // グローバルオブジェクトの生成
l->new (); // ローカルオブジェクトの生成
```

Execution Model of OZ++

NIIBE Yutaka (Mitsubishi Research Institute, Inc.),
TASHIRO Shuichi (Electrotechnical Laboratory),
SUZUKI Takayuki (Sharp Business Computer Software, Co., Ltd.*),

HIRAKAWA Hidetada (Nihon Unisys, Ltd. *),
TSUKAMOTO Michiharu (Electrotechnical Laboratory)

*: Researcher of Research, Development and Evaluation of Fundamental Software Technology in Information-Technology Promotion Agency, Japan

3 オブジェクトのコピーイン/アウト

ローカルオブジェクトは、グローバルオブジェクト間で共有されることはない。グローバルオブジェクト間でローカルオブジェクトが渡される時、そのローカルオブジェクトはコピーされる。

グローバルオブジェクトのメソッド起動時、引数のローカルオブジェクトは呼び出し側でコピーアウトされ、呼び出された側にコピーインされる。同様にこのメソッドの終了時、返り値のローカルオブジェクトはコピーアウト/コピーインされる。

4 実行モデル

図1にOZ++における実行モデルの図を示す。

グローバルオブジェクトはネットワーク上でサービスを提供する単位である。グローバルオブジェクトのメソッドの実行はネットワークサービスの実行である。ローカルオブジェクトはグローバルオブジェクトの部品となり、メソッドの起動時の引数として、および結果の値としてネットワークを渡る。

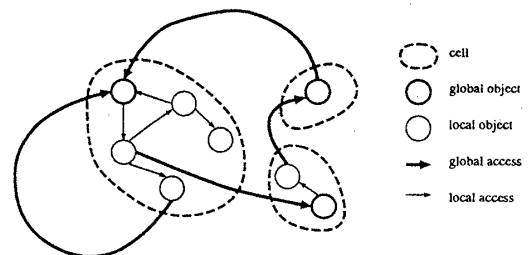


図 1: 実行モデル

5 スレッドの並列実行と排他制御/同期

OZ++では同一のオブジェクトに対し、複数のメソッドが並列実行され得る。スレッドの生成はグローバルオブジェクトのメソッド起動時に起こる。ローカルオブジェクトのメソッドの実行は、スレッドの生成を伴わずに手続き呼出しで実行される。

スレッド間の排他制御を行なうオブジェクトはモニタである。モニタのオブジェクトでは、各メソッドはロックを得るか否かの属性を持つ。ロックを必要とするメソッドは排他的で、同時に複数のスレッドで実行されることはない。スレッド間の同期はオブジェクトのインスタンス変数の条件変数によって行なう。

```
Condition item, space;
int producer(): locked; // 排他制御メソッド
{
    while (/* buffer is full */)
        wait space;
    // insert item to buffer
    signal item;
    // ...
}
int consumer(): locked; // 排他制御メソッド
{
    while (/* buffer is empty */)
        wait item;
    // delete item from buffer
    signal space;
    // ...
}
```

6 プロセスとオブジェクト

OZ++ではメソッド呼出しによるスレッドの連鎖をプロセスと呼び、制御の流れの抽象としてこれを扱う。

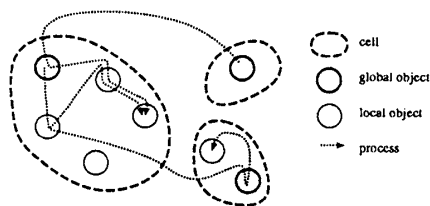


図 2: プロセスとオブジェクト

メソッド起動時にフォークすることを指定すると、新たなプロセスが生成され、元のプロセスと並行に実行が進む。プロセスの生成は呼出し側の制御による。

プロセスに対しては、終了を待ち結果を得る join, 分離する detach, 実行を中断する abort などの操作が用意されている。

```
int @p; // int を返すプロセス p
int i;
Aclass a;
    // 非同期実行, プロセスの生成
p = &a->method(args);
...
i = @p; // プロセスの join, 値の受けとり
```

7 メソッドの動的な束縛(多相性)

OZ++言語において変数は静的な型(クラス)を持つ[2]が、その変数にはそのクラスの子孫のオブジェクトが代入され得る。これにより、変数は複数のクラスのインスタンスを参照する多相性を持つ。実行されるメソッドはオブジェクトの実行時の型によって決まる。

8 おわりに

オブジェクト指向分散環境 OZ++の実行モデルについて、グローバル/ローカルオブジェクト、メソッドの並列実行とプロセスについて述べた。

OZ++ではこのモデルにより、資源の共有、プロセスの並列実行を定義し、分散環境に適応させたオブジェクト指向プログラミングを可能としている。

本研究は、情報処理振興事業協会「開放型基盤ソフトウェア研究開発評価事業」の一環として行われたものである。関係各位に感謝する。

参考文献

- [1] 塚本他: 「オブジェクト指向開放型分散システム OZ+の研究開発」, 電総研彙報, vol. 56, No. 9, Sep. 1992
- [2] 西岡他: 「オブジェクト指向分散環境 OZ++ の言語の基本設計」, 情報処理学会第46回全国大会, Mar. 1993
- [3] Trehan et al.: "Concurrent Object Oriented 'C' (cooC)", ACM SIGPLAN Notices, Volume 28, No2, February 1993
- [4] 濱崎他: 「オブジェクト指向分散環境 OZ++ の基本設計」, SWoPP '93, Aug. 1993