

システム統合化をめざしたLAN (UNINET)

3T-3

その3. ソフトウェア分割作成システムとしての評価

相原玲二 山田英司 間伸一† 山下雅史 阿江忠
(広島大学 工学部 †現在 NTT)

1. まえがき

モジュール化プログラミングはプログラムを部品化することでシステム設計、作成、デバッグ、保守を容易にする。特に多人数で大規模プログラムの開発をする場合、LAN環境下の各ワークステーションにおいて独立に作成、デバッグを行ないながら目的のプログラムを構築すれば、効率のよいソフトウェア開発が期待できる。

今回取り上げたプログラミング言語Modula-2は、モジュール化プログラミングを行うためのモジュール機能を持ち、さらにモジュール単位で分割コンパイルが可能である。本稿では、このModula-2でのプログラム分割作成についての考察およびLAN上で実現した場合の評価について述べる。

2. Modula-2のモジュール

Modula-2のモジュールには

- ・プログラムモジュール
- ・定義モジュール
- ・実現モジュール

の3つがある^[1]。プログラムモジュールはPascalのprogramに当たる。Modula-2でプログラムをモジュール化するとき、1つのモジュールを1組の定義モジュールと実現モジュールを用いて記述する。定義モジュールには、他のモジュールに開放(EXPORT)する(言い替えると、他のモジュールが導入(IMPORT)する)オブジェクト(定数、型、変数、手続き)の定義などが書かれる。実現モジュールでは、対応する定義モジュールで定義されたオブジェクトの具体的な記述が行われる。この2つのモジュールからライブラリが構成される。

Modula-2のコンパイルはモジュール単位で行われる。しかも、分割コンパイルが可能である。これは独立コンパイルと異なりモジュールにわたって導入されるオブジェクトの型チェックが行われる。

Modula-2のコンパイル順序には制約があり、

- (1) 定義モジュールは対応する実現モジュールよりも先にコンパイルされている。
- (2) あるモジュールが他の定義モジュールから開放

されるオブジェクトを導入するとき、定義モジュールは先にコンパイルされている。

という2つの条件を満たしていなければならない^[2]。

3. モジュール化の例

次に、多人数でプログラミングすることを考慮したモジュール化の実用的な方法について説明する。この方法ではまず、プログラム全体をプログラムの持つ機能を単位としておおまかにモジュール化する。しかし、このままでは、いくつかのモジュールが同じオブジェクト(変数や手続きなど)を使用している場合があり、もしこのオブジェクトをどれかのモジュール中に置くと、モジュール間に開放、導入の関係ができ、それぞれのモジュール(プログラムモジュールを除く)が独立したものとならなくなる。そこで、これらのモジュールから共通して使われるオブジェクトをまとめて新しい1つのモジュールにする。実際のプログラムについて、このモジュール化を行った例が図1である。

このようにモジュール化すると、プログラムモジュール以外のモジュールを書き替えることなく新しいモジュール(新しい機能)を簡単に追加することができる。

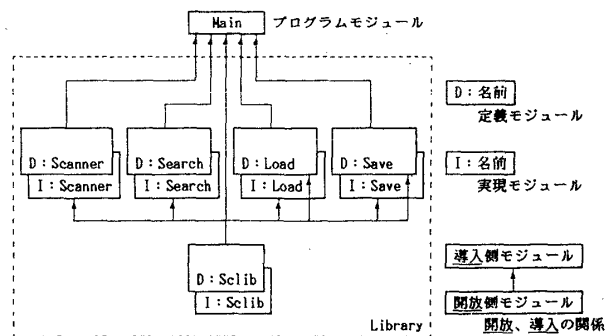


図1 モジュール化の例

4. LANによる分割コンパイルの実験

実験は、複数のワークステーションにLANを使ってファイルを送り、コンパイルを行わせ、結果を再び

LANを使って回収する、というものである。実験に必要なLANの基本機能は、ファイル転送とリモートプロセス起動である。

今回使用したLANシステムは、既に稼働中のLAN^[3]のデータリンク層以下の機能を利用したネットワークユーティリティで、これはファイル管理をサポートし、リモートプロセス起動機能を持っている。以下では図2で示されたネットワークユーティリティの各部(上位3層)について簡単に説明する。

- 1) コマンド/ユーティリティ関数部では、ファイルマネージャ部の基本機能関数を用いてコマンド及びライブラリ関数を用意している。
- 2) ファイルマネージャ部では、ネットワークを介したファイル操作のための基本的機能としてファイルのオープン、クローズ、リード、ライト、キル、ディレクトリ参照、リモートプロセス起動機能、及びメッセージライト機能を提供している。
- 3) 簡易トランスポート部では、相手アドレス管理、データの送受信機能を提供している。

本システムのファイル転送コマンドを用いた結果、その速度は約65kbpsであった。(ディスク装置としてメモリ上に仮想的に作られたものを使用した。)

ネットワークユーティリティ

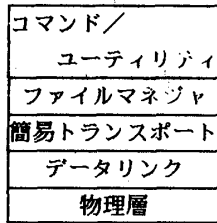


図2 プロトコル階層とネットワークユーティリティ

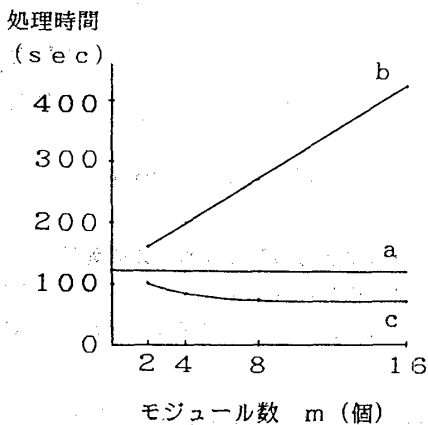


図3 非分割と分割コンパイル

5. LANを用いたプログラム開発例

実験は、プログラムモジュール以外のモジュール間には開放、導入の関係がないモジュールからなるプログラムのコンパイルについて行った。

- 1) a. 非分割コンパイルした場合
b. 1台のワークステーション上で分割コンパイルした場合
c. LANに接続された複数台のワークステーションを使用した場合
- について、それぞれの処理時間を測定した(図3)。
- 2) 1)の結果をもとに、分割コンパイルのLANシステム上へのインプリメントを検討した(図4)。図4中の65kbpsのグラフは、我々のLANのネットワークユーティリティを用いた時の予想値である。

6. むすび

LANを用いるソフトウェア分割作成システムについての一例を述べた。ハードウェア的には現状のLANでも十分実用になることが判明した。

参考文献

[1] N. Wirth, Programming in Modula-2, Springer-Verlag, 1983(斉藤訳, Modula-2プログラミング, 日本コンピュータ協会).

[2] R. Gleaves, Modula-2 for Pascal Programmers, Springer-Verlag, 1984(阿江訳, Modula-2 - Pascalからの入門-, 啓学出版).

[3] 新田 他, "パーソナル・コンピュータ・ネットワークOSのためのプロトコル階層," 信学技報, IN85-78, 1985.

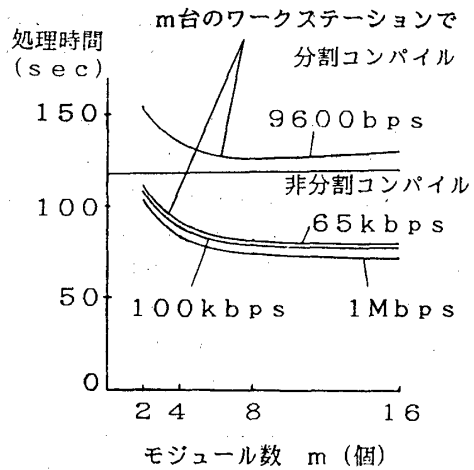


図4 LANの速度による違い