

1 T-9

LOTOS実行系の実装結果

野村真吾 瀧塚孝志 長谷川亨  
国際電信電話株式会社 研究所

1. はじめに

LOTOS<sup>[1]</sup>は、通信プロトコルの仕様を記述するために、ISOで標準化された仕様記述言語の一つである。筆者は、LOTOSで記述された仕様を、C言語のプログラムに変換して実行するLOTOS実行系を作成している<sup>[2]</sup>。本稿では、実行系のプログラムサイズ、およびプロセス生成とプロセス間通信の実行時間、さらにLOTOSによる小規模な仕様を変換した結果について報告する。

2. LOTOS実行系

2.1 LOTOS

LOTOSでは、ひとまとまりの動作をプロセスとして記述し、動作の結合オペレータを用いてプロセスの並列動作関係を記述する。プロセスは他のプロセスと情報交換する窓口(ゲート)を持つ。プロセスの動作仕様は、ゲートにおいて観測されるイベントの時間的順序を列挙することにより規定される。

LOTOSによる仕様は、以下の特徴を持つ。

- ① 手続的ではなく、宣言的に記述を行うため、プロセス数が多く、並列度も高い。
- ② 既存のOSおよび並列言語と異なり、1度に複数のプロセスが同期し、同期時に値の送受が双方向で行われ送信者と受信者の区別がない。

2.2 LOTOS実行系

本実行系は、トランスレータとスケジューラにより構成される(図1)。トランスレータは、LOTOSで記述されたプロトコル仕様を、スケジューラの機能を使用するC言語のプログラムに変換する。スケジューラは、プロセスの実行やイベントによる同期と値の送受等、LOTOSの言語仕様で規定される基本的な動作を提供する。

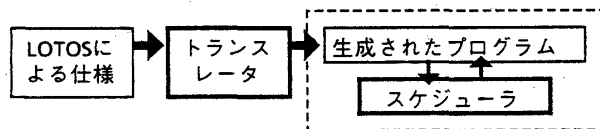


図1 LOTOS実行系の構成

3. LOTOS実行系の実装結果

本実行系の実装および実験は、VAX Station-3100 (VAX780を1MIPSとして2.8MIPS)上で行った。

3.1 プログラムサイズ

トランスレータおよびスケジューラのプログラムサイズは、Cのソースコードでそれぞれ約9.6k行と約2.3k行であった。プロセス管理の一部は、アセンブラで約100ステップ記述した。

3.2 実行時間の測定

(1) プロセス生成

多くのプロセスが存在し、頻繁に生成/消滅が繰り返されるLOTOS仕様を実行するためには、同時に存在可能なプロセスの個数とプロセス生成時間が問題となる。

各プロセスの作業領域をAdaにおいてタスクに割当てられる作業領域と同じ10kBとし、標準的なシステムの環境設定でプロセスを生成したところ、約1,400個のプロセスが生成できた。このとき1プロセスあたりの生成時間は、約600 $\mu$ sであった。

本実行系では、プロセス毎に作業領域の大きさを変えることも可能である。

(2) 同期するプロセスの数と実行時間

1度に同期するプロセスの個数を変えて、1番目のプロセスが同期の要求を出してから同期が完了するまでの時間(同期処理時間)を測定した(図2)。

処理時間は、10万回の繰り返し時間を割って求めている。

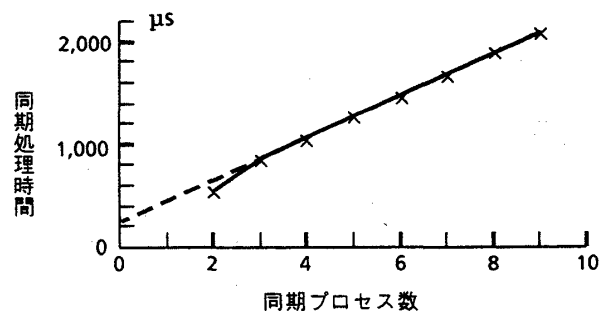


図2 同期プロセス数と同期処理時間

### (3) イベントの宣言数と実行時間

イベントは、一つのゲート識別子と複数個の値宣言または変数宣言の並びにより定義される。そこで、一つのイベント内で定義される宣言の個数を変えて、2つのプロセスの同期処理時間を測定した(図3)。両プロセスとも値宣言による同期である。

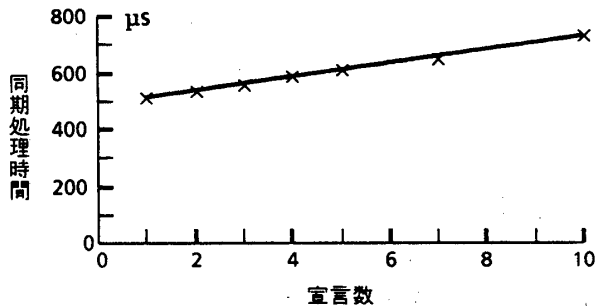


図3 イベントの宣言数と同期処理時間

### 3.3 プログラム変換結果

特別なデータ(優先データ)の追越しを考慮したFIFOキューのLOTOSによる仕様<sup>[1]</sup>を、トランスレータによりCプログラムに変換した。変換したプログラムにデータ型の定義、外部プログラムとの入出力処理を追加し、実行可能なプログラムを作成した(図4)。

作成したプログラムに、値の入力プログラムと出力プログラムを加え、図5のように接続して、動作を確認した。

入力仕様 (LOTOS)	変換後プログラム	記述追加部分	FIFOキュープログラム
29行	239行	66行	305行

図4 コード量

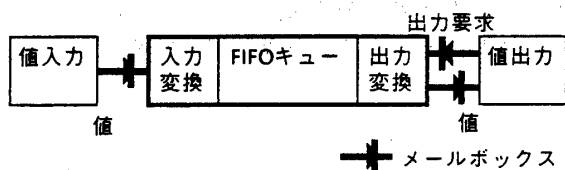


図5 プログラムの構成

### 4. 考察

(1) トランスポートプロトコルの仕様<sup>[3]</sup>では、68個のプロセスが定義されている。同一プロセスが複数生成されることがあり、仕様実行中に生成されるプロセス数は数百近くになると考えられる。本実行系では、標準設定のシステム上で1,400個のプロセスを生成することが可能であったので、通信システムの構築に耐えられると思う。

ユーザのメモリ割当て量を増加させれば、さらに多くのプロセスの生成が可能になる。

(2) 同期処理時間の測定結果から、2つのプロセスが同期する最も単純な場合に必要時間は、約500μs

である。

VAX VMSの提供するメールボックスによるプログラム間の同期通信に要した時間は、720μsであった。メールボックスによる同期通信は、2つのプログラム間で情報の送信/受信を特定したものである。送受信を同時に行うLOTOSの同期通信を実現するためには、さらに多くの時間が必要となる。

従って、LOTOSのプロセスをOS上のプロセスに対応させる方法に比べ、本実行系は効率的な同期処理を実現している。

(3) プロセスの同期処理が、プロセスを待たせる処理、イベントの各宣言をマッチングする処理、プロセスのスケジューリング処理、およびこれら以外の処理からなるとする。3.2の実験結果から、それぞれの処理時間は、49μs、25μs、134μs、218μsと求められた。

文献[3]から、68個のプロセスが、宣言数3のイベントを4回発行して、トランスポートのコネクションを確立すると仮定する。本処理系を用いてトランスポートプロトコルプログラムを作成した場合、1つのコネクション確立に含まれる同期処理時間は、約70msになる。

(4) 小規模ではあるが、FIFOキューの仕様を用いてプログラムが作成できることを確認した。外部プログラムとの入出力や、内部的なデータ構造と外部のデータ構造の変換を行うプログラムは、LOTOSプロセスとして扱われるよう記述した。

このプロセスにおいて、同期型の処理を用いるとプログラム全体が停止してしまう。そのため、非同期の通信を用いて値の送受を確認する機能をスケジューラに持たせた。

### 5. おわりに

LOTOS実行系の実装結果について述べた。多くのプロセスを生成することが可能で、効率の良い同期処理が実現されていた。また、本実行系を用いたプログラム開発が可能であることも確認した。今後、大規模な通信プログラムに適用するとともに、さらに実行効率を向上させる方法を検討していく予定である。最後に、日頃御指導頂くKDD研究所小野所長、山田次長、通信ソフトウェア研究室小西室長に感謝します。

### 参考文献

- [1]: ISO 8807, "LOTOS - A formal description technique based on the temporal ordering of observational behaviour", Feb. 1989.
- [2]: S.Nomura, T.Hasegawa, T.Takizuka, "A LOTOS Compiler and Process Synchronization Manager", Protocol Specification, Testing and Verification, X, IFIP, 1990.
- [3]: DTR10167, "Guidelines for the Application of Estelle, LOTOS, and SDL", Jan. 1990.