分散処理用言語 DIPROLの実装方式

7H-6

小森 育　田中英彦　元岡 達
(東京大学 工学部)

はじめに DIPROL[1]はメッセージ通信に基づく分散システム用の並行処理記述言語である。記述の対象となる複数のプロセスは、共有メモリを持たず、それぞれメッセージ通信のためのインターフェイスとして、ポートと呼ぶ実体を複数もてる。記述されるプロセス群は、それぞれのポートを接続することで一つに構成する。分散したシステムにおける通信の伝搬遅延を考慮し、通信を行なうためのポートと、それが属するプロセスの並行動作を許している。さらに、複数ポートに対する非決定性コマンドが容易に実装できるようにしてある。プロセスのポートの使い方の重要な性格のいくつかについて、宣言を積極的に義務づけ、ポートの接続に際して、整合性のチェックをするようにしている。以上のような特徴をもつ言語DIPROLの実装について検討したので、以下に報告する。

二階層の仕様

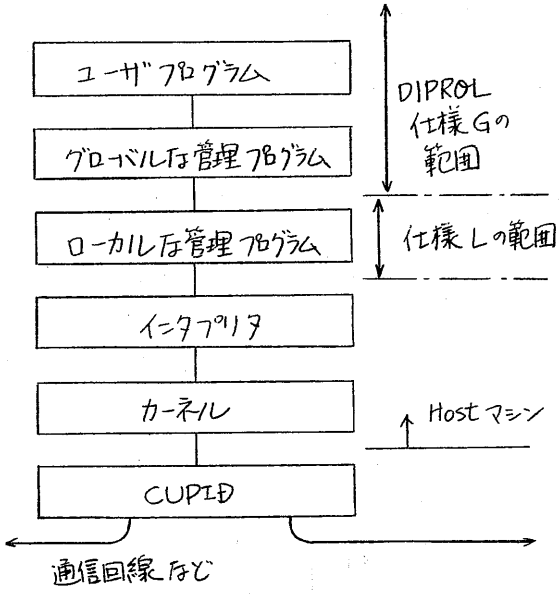
実装に際して、DIPROLの仕様を二層に分ける。一つは分散システムの個々の要素内の資源の管理を記述するための仕様Lであり、いま一つは、前者のローカルな資源管理の機能を仮定して、分散したシステム全体にわたる機能を記述し、実現するための仕様Gである。地理的に分散したネットワークを対象とした場合、各ホストごとのローカルな資源の管理を仕様Lで、ネットワーク上での管理を仕様Gで実現する方針をとる。

実装の概観

実装を進めつつある対象は、本研究室において開発したプロセス間通信用の専用プロセッサCUPID[2]をそれぞれ装備したミニコン2台であって、機能分散をめざした実装が試みられる。

CUPIDは通信制御用FEPの役割りを含む上に、ローカル/リモートの差なくすべてのプロセス間通信(IPC)を扱うことができるように設計されている。

DIPROLのコンパイラはPascal-Pのコンパイラを利用して作られており、実行はインタプリタを必要とする。分散システムの実装の全体的な構造は右図のようになる。



カーネル

カーネルに仮定する機能は

1、プロセス間通信(IPC)の機構 (非決定性のIPCコマンドを含む)
2、割り込み機構のサポート (DIPROLでは割り込みの記述もする。)
そしてもしあれば、)
3、プロセスディスパッチャとスケジューラの基本メカニズム
4、(仮想)記憶管理の基本部分

といったものであり、基礎的なメカニズムをカーネルに仮定し、アルゴリズム、ポリシーと言える部分は、なるべく、DIPROLで記述するように考えている。

IPCのためのカーネルコール　カーネルは IPCのためのポートを、カーネルオブジェクトとして管理する。ポートに関するカーネルコールは大きく二つに分けられる。一方は、ポートをつくり出したり、ポート間の静的な接続状態を操作するためのものである。ポートの管理はこれらを用いて構成する。もう一方は、実際のメッセージごとのIPCを操作するためのポートコマンドである。これらは、単独で用いる他に、複数ポートにまたがる非決定性コマンドの一項目として使うことができる。(下図)

静的なポート管理のためのカーネルコール
makeport
freeport
connect
disconnect
select
who　など

コマンド形式	単一コマンドとしての意味	非決定性コマンド時の選択条件
CALL	ポートを起動後通信終了待ち。	通信開始可
START	ポートを起動するのみで待たない。	通信開始可
MATCH	ポートが通信可になるまで待つ。	通信開始可
WAIT	起動ずみのポートの通信終了を待つ。	通信終了
ABORT	起動ずみのポートの通信を中断する。	――
DELAY*	タイマーが尽きるまで待つ。	タイマが尽きたこと

* DELAYはポートコマンドではない。

非決定性のポートコマンド　非決定性のポートコマンドでは、複数の項目中から一つを選んだ後は、その単一コマンドを実行する。DIPROLでは非決定性のコマンドは送信/受信ともに含むことができるが、分散システムを意識し、相互に干渉する非決定性コマンドが通信のオーバヘッドを過度にひきおこさないように[3]、ポートの性格づけをきめ細かくしてある。まず、プロセスが、非決定的に選択実行する通信のために使うことのあるポートは passive という属性で宣言し、このポートのパートナーとなるべきポートは、必ずactive属性で宣言されていなければならない。(コンパイラ等は、activeなポートが単一のパートナーしかもてないこと、そして非決定性コマンドの一項目として使われないことを監視する。)こうすれば、passiveなポート同士の接続を発見できるので、非決定性コマンドの選択動作は局所性を保ってインプリメントできる。

CUPIDとの機能分担　現在のCUPIDは、非決定性のコマンドとして1つのポートが、複数のパートナーポートのいずれとも通信できる RECEIVE-ANY / -SELECTIVE をサポートしているが、複数ポートにまたがった非決定性はサポートしていない。そこで、とりあえずホストマシンの機能として複数ポートの非決定性を実現することにした。その結果、ポートの状態管理の制御ブロックはホスト内におかれる。これは、マルチプログラミングの立場からは、IPCによるプロセス統合の時点が直接使える利点があるが、インターフェイスの切り口としては、プロセス統合のメカニズムも、CUPIDに移した方が美しい。

割り込み処理　DIPROLは割り込みをポートとみたてて扱う記法をとれる。しかし、実装上、I/O割り込みはカーネルが、直接、担当のプロセスへ制御を渡すことにし、他のIPCのようにCUPIDを介することはしない。割り込み(を扱う)プロセスの走行レベルをすっきりした形で実装するためには、割り込み原因ごとのマスクや、ソフトウェアで起こす割り込みが使えることが望ましい。

ポート間リンクの管理

DIPROLはバーチャルコール形のIPC管理を前提にしている他、プロセス間でのパートナープロセスの終了などの例外もポート間リンクを通じて実現する。この管理を仕様レベルのDIPROL自身で記述するときには、前述の静的なポート管理のためのカーネルコールを用いるわけである。

この他、プロセス管理のためのカーネルコールと、メモリ管理のためのカーネルコールのインターフェースを設定したが、仮想記憶を用いるシステムでは、そのための一般的メカニズムを特定のポリシーから独立させることが難かしく、カーネルの役割りはシステムに依存して変化する。

参考文献　[1] 、小森、田中、元岡、「分散処理用記述言語 DIPROLの検討と実装」　情処第23回全国大会, 4D-10

[2] 和賀井、田中、元岡「網向きプロセス間通信制御プロセッサCUPIDの実現と評価」情処第23回全国大会、5D-8

[3] Mao, T.W and Yeh, R.T "Communication Port: A Language Concept for Concurrent Programming" IEEE Trans. Software Engeneering Vol. SE-6, No.2 pp194-204 (Mar. 1980)