

分散システム記述言語DIPROLの 支援環境

6G-6

赤田 正雄 田中 英彦 元岡 達

(東京大学 工学部)

1. はじめに

DIPROL [1] は、当研究室で設計されたメッセージ通信に基づく分散システム用並行処理記述言語であり、対象とする分散システムは、①通信コストが無視できない、②共有メモリを仮定しない、計算機網(広域、ローカルを問わない)である。

DIPROLのプログラムを開発・実行する環境は、一般の言語と異なり複数の計算機上にまたがることになり、ネットワークを構成する各マシンごとにad hocな支援環境ではなく、ネットワーク全体に統一的な支援環境を設定しなければならない。

現在、プロセス間通信用の専用プロセッサCUPID [2] を介して、通信回線で結ばれた2台のミニコン(PANAFACOM U300, OKITAC 4300C)に対するDIPROLの実装が進められており、本論文では、東大大型計算機センタのM280H上に実装中の、DIPROLのコンパイラ/ポート・リンカ/アセンブラも含めたDIPROLの支援環境について考察を行なったので、その報告を行なう。

2. DIPROLの支援環境

現在のところ、DIPROLはNOS(Network Operating System)記述用の言語という仕様になっている。従って、NOSが構成され、ネットワーク中の資源に対する統一的なアクセス法が提供されるようになると、新たなユーザ・インターフェースをもった、新たなDIPROLの仕様が必要となる。NOS記述用をDIPROL/S、ユーザ・アプリケーション用をDIPROL/Uと呼ぶ。このそれぞれに対して、支援環境DSSE(DIPROL/S Support Environment)とDUSE(DIPROL/U Support Environment)が、設定される。以下、この内DSSEについて述べる。

DSSEは、DIPROL/SによってNOSを記述し、それを走らせるための支援環境である。その内容は、コンパイラ、ポート・リンカ、etc.で構成されるDIPROLのプログラムの開発時支援と、ネットワークを構成する各マシンに最低限必要とされる、実行時支援環境KDSE(Kernel DIPROL Support Environment)から成る。

3. 開発時支援

①コンパイラ

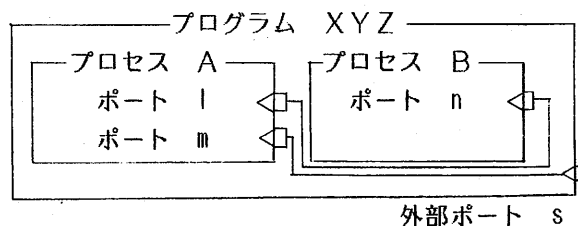
DIPROLのコンパイラは、PASCAL-Pのコンパイラ(Pascalで記述されている)を改造して、M280H上に実装されており、仮想スタック・マシンのアセンブリ言語をオブジェクトとして、出力するようになっている。

コンパイル単位はプログラムと呼ばれ、ポートと呼ばれるインターフェースを介して通信をしようとするプロセスから構成される。一つのプログラム内には、複数のホストで動くプロセスを混在して記述できる。

DIPROLのコンパイラは、プロセスごとに、各マシンのデータ表現に合せたコードを出力する。従って、コンパイラは、ホスト名で定まる各マシンに関する情報を、プロセスの属性としてもっている必要がある。

プロセス間の通信のインターフェースであるポートは、非常に強く型付されており、コンパイラは、コンパイル単位内でCONNECT文により接続されているポート間の整合性のチェックを行なう。又、実行時にポートを動的に結合することが可能なように、外部ポートだけでなく、全てのポートの属性が、オブジェクト・コードと一緒に出力される。

fig. 1 DIPROLのコンパイル単位



Program XYZ

外部ポートの宣言

```

process A on U300 ;
process B on OKI ;
connect A. l >> B. n ;
export s = A. m
end .
    
```

②ポート・リンカ

ポート・リンカは、一般のリンカとは異なり、コンパイラのオブジェクトそのものには関与せず、各コンパイル単位の外部ポート間の接続の整合性のチェックを行なう。従って、ポート・リンカは出力コンパイラの出力と同じ形式の仮想スタック・マシンのアセンブリ言語である。ポート・リンカもM280H上に実装される。

③アセンブラ

アセンブラは、仮想スタック・マシンのアセンブリ言語を各マシンのインタプリタが解釈できる形にアセンブルする。アセンブリ言語の段階では、一つのプログラムの中に、複数のマシン上にロードされるプロセスが混在しているが、アセンブルされた結果は、各マシンごとにまとめられた形になる。アセンブラもM280H上に実装される。

4. 実行時支援環境—KDSE—

KDSEは、イニシャル・ローダ、各マシン用のインタプリタ、カーネル、そしてCUPIDから構成される。この内、カーネルについて述べる。

DIPROLでは、基本的な資源管理のアルゴリズムは、DIPROL自身で記述されるので、カーネルは、そのために必要な基本的な手段をカーネル・コールの形で提供する。KDSEのカーネルとして必要な機能には次のようなものがある。

①割込処理

ハードウェア異常割込、スーパー・バイザ割込、プログラム・チェック割込などに対しては、カーネル内で処理をするが、入出力機器からの外部割込は、ポートに対する通信と同様に扱うような機構をサポートしている。従って、外部割込処理は、DIPROL自身のINTERRUPT文を用いて記述される。

②プロセス間通信の機構

リモート/ローカルを問わず、プロセス間通信は、カーネルがCUPIDに処理を依頼することによって行なわれるが、ポートを実現しているのは、カーネル自身である。あるポートが、現在どのポートと接続して、どのような状態にあるかという情報は、カーネル内に保持されている。ポートの状態遷移の要因となるのは、プロセスからのカーネル・コールとCUPIDからの割込、タイマからの割込である。

また、DIPROLでは、プロセスとポートの並行動作を可能にするために、ポートにバッファを設けることができる。このバッファの管理を行なうのもカーネルの仕事であり、リモート通信で送られてきたメッセージのデータ表現の変換も行なう。さら

に、DIPROLの非決定性の構文SELECT文をサポートするのもカーネルが行なう。

③入出力機構

プロセス間の入出力はポートで行なうが、周辺I/Oとの入出力は、DIPROLの仕様の範囲ではない。カーネル・コールI/Oを提供するが、そのパラメータのもつ意味は各マシン、デバイスに依存する。

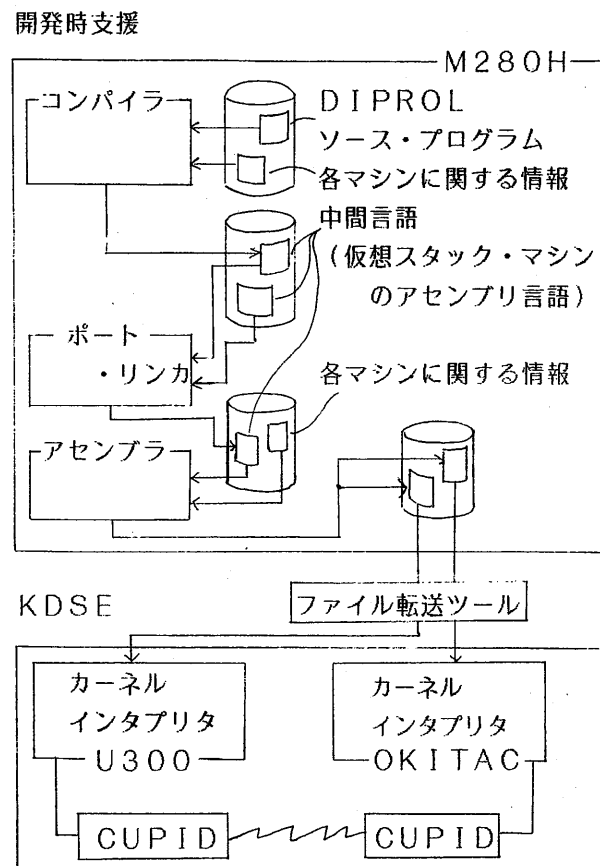
④その他

メモリ管理、プロセス管理etc. のためのカーネル・コールの提供。

5. おわりに

以上のDSSEのもとにNOSが構成される。DUSEは、NOSの構成をいかにすべきかという一つのテーマを含んでおり、DSSEの実現後の課題である。

fig. 2 DSSEの構成



<参考文献>

- [1]. 小森, 他. 「分散処理記述言語DIPROLの実装方式」、情処24全大、7H-6.
- [2]. 和賀井, 他. 「網向きプロセス間通信制御プロセッサCUPIDの実装と評価」、情処23全大、5D-8.