5K-6

分散処理向き高級言語 N-PASCALの検討

堀　健一、松方　純、和賀井フミ子、
田中英彦、元岡　達（東京大学工学部）

1、はじめに

計算機網上に分散処理システムを構成するための高級言語、N-Pascal を試作、実装したので報告する。

N-Pascal は次の特徴を持っている。

① プログラムは並行に動作するプロセスの集合として記述される。

② プロセス間の通信機構は、メッセージの授受という方式で統一されている。

ここでは、N-Pascal の基本的仕様、実装方法を述べる。

2、並行動作プロセス

N-Pascal のプログラムは、特定のホスト計算機上で並行に走るプロセスの集合から成る。　各プロセスは、従来の Pascal のプログラムとほぼ同じものである。

```
PROGRAM system = (pa,pb @u300);
EXTERNAL (pc,pd @pps1),
         (pe,pf @oki);

PROCESS pa;
 CONST . . .
 TYPE . . .
 VAR . . .
  PROCEDURE . . .
     :
    send(message,param,'L0');
     :
 END
ENDPROC;
PROCESS pb;
     :
     :
    receive(buffer,param,'L1');
     :
     :
ENDPROC
ENDPROG.
```

図1　N-Pascal プログラムの例

N-Pascal による分散処理システムは、網を構成する各ホスト計算機に、別々にコンパイルしたプログラムをロードして走らせることにより、互いに通信をし合うプロセス群として実現する。

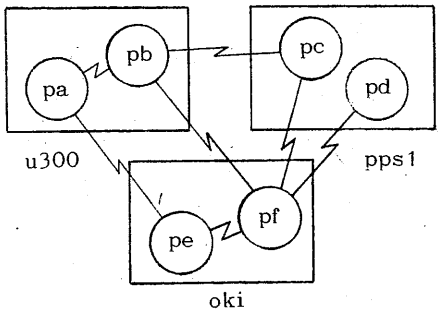

図2　分散処理システム

3、プロセス間通信機構

プロセス間の通信は、通信相手のプロセスが同一計算機上にあろうと、異なる計算機上にあろうと、メッセージの直接的な授受による。　これによって、網内に散在する各種の資源には、同じ形式、方法でアクセスすることが可能となる。

プロセス間通信のために、以下の2つの標準手続きを用意した。

```
TYPE comparam = RECORD
                  status:comresult;
                  modifier:options '
                END;
     label = ARRAY(.1..2.)OF char;

PROCEDURE send(buffer:anytype;
               VAR p:comparam;
               l:label);

PROCEDURE receive(VAR buffer:anytype;
                  VAR p:comparam;
                  l:label);
```

ここで、buffer は任意の型の変数である。　l は通信相手を指定するためのラベルである。

図３に示すように、 send を提出したプロセスと、 receive を提出したプロセスの要求が一致してから、実際のデータ転送は行なわれる。

特に、 p.modifier := [nonblock] として、nonblock モードの通信要求を行なうと、データ転送が終了しないうちにプロセスの実行を再開することができる。

receive については

p.modifier := [any] として、自分宛のメッセージはすべて受け取るという receive-any の機能を設けた。

他にも、通信を補助するための標準手続きをいくつか用意した。

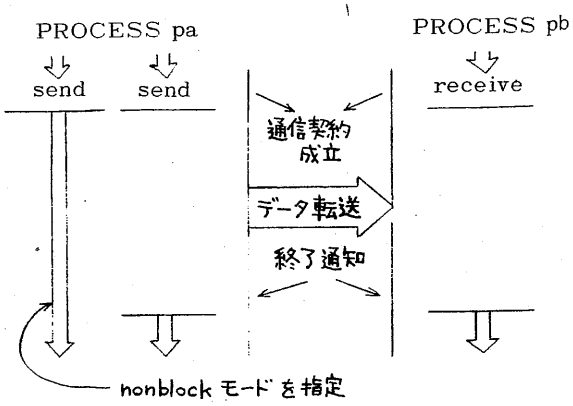

図３　send と receive

4、プロセスの識別

各プロセスは、他のプロセスを実行時に明確に識別するため、ラベルを用いる。

コンパイル時に名前のわかっている他のプロセスに対しては、標準手続き regist によって、ラベルとプロセスの関係を登録することができる。

また、 receive-any 要求によって通信契約が成立した時にも、ラベルと相手プロセスとの関係が登録される。

```
regist(pps1reader,'L0');
      あるいは
p.modifier:=(.any.);
receive(buffer,p,'L1');
```

図４　ラベルの登録

5、N-Pascal の実装

N-Pascal は既在の並行処理システム記述言語 Concurrent Pascal (C-Pascal) を利用して実装した。 C-Pascal は、Sequential Pascal (S-Pascal、従来の Pascal とほぼ同じもの) で記述されたユーザ・プログラムを実行することができる。

N-Pascal のソース・プログラムは、プリ・プロセッサによって、各プロセスに対応した S-Pascal プログラムに変換される。 これらは C-Pascal で記述されたプログラム (N-Kernel) 上で並行に実行される。

プロセス間通信の処理は、すべて、外付の通信制御プロセッサ (CCP) に依頼して実行する。

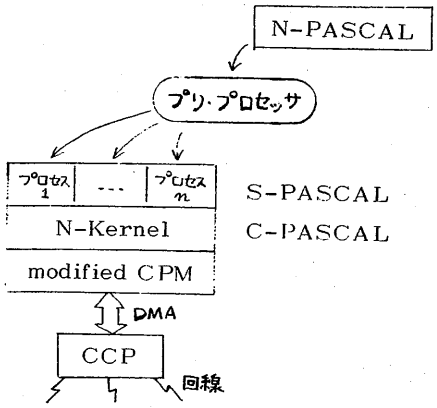

図５　N-Pascal システム

6、今後の課題

現在のところ、本言語でまとまった大きさのシステムを記述するには至っていない。 今後は、このような作業を通して言語の記述能力、システムとしてのスループット等を評価し、また、エラーからの回復、デッドロックの問題等の検討を重ねたい。

さらに、プリ・プロセッサ方式ではない、コンパイラの作製も課題である。

参考文献

金田裕之他「プロセス間通信の機能を備えた通信制御プロセッサ」本予稿集 5k4。