

並列計算機「砂丘」のためのプログラム開発システムについて

4B-9

鈴木俊正 井上倫夫 小林康浩
鳥取大学工学部

1. はじめに

効率よく並列処理を実行させるためのプログラミングを作成するには、多くの時間と手間がかかる。そこで今回、並列計算機のためのプログラムを容易に作成できるようにするためのシステムを開発したので報告する。

2. 全体のシステム構成

我々の研究室では資源共有型並列計算機「砂丘」を開発しており、64台の68000PUを使用した並列処理を行わせることができる。今回のシステムは、この「砂丘」で動作する並列処理プログラムを組むことを目的としている。

プログラムの製作過程は、ソースプログラムの作成から実行ファイルへの変換まですべてUNIXワークステーション上で行うようにしている。利用者は画面上に表示された選択肢の中から処理内容を選択しながらプログラミングできる。作成された実行ファイルは「砂丘」に転送され、実行される。

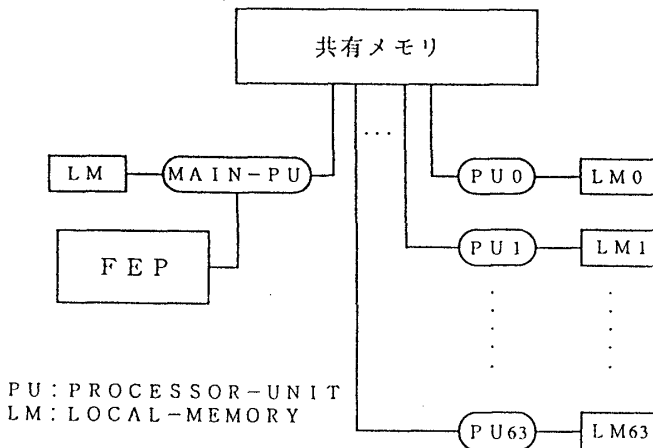


図1. 利用者から見た「砂丘」の構成

3. ソースリストの作成方法

ソースリストはPADと呼ばれるプログラム表記法を用いて画面上に表示される。ここでは並列計算機のためのプログラムを作成するので、一般のPADの表記法に独自の表記法をいくつか追加したものを使用している。ソースプログラムの作成は、箱の種類選択、箱の位置の選択、箱の内容の選択の3つの処理を繰り返すことによって行えるので、目的の処理のアルゴリズムさえ正確にわかっているならば比較的簡単に目的のプログラムを作成することができる。下図にプログラムの例を示す。

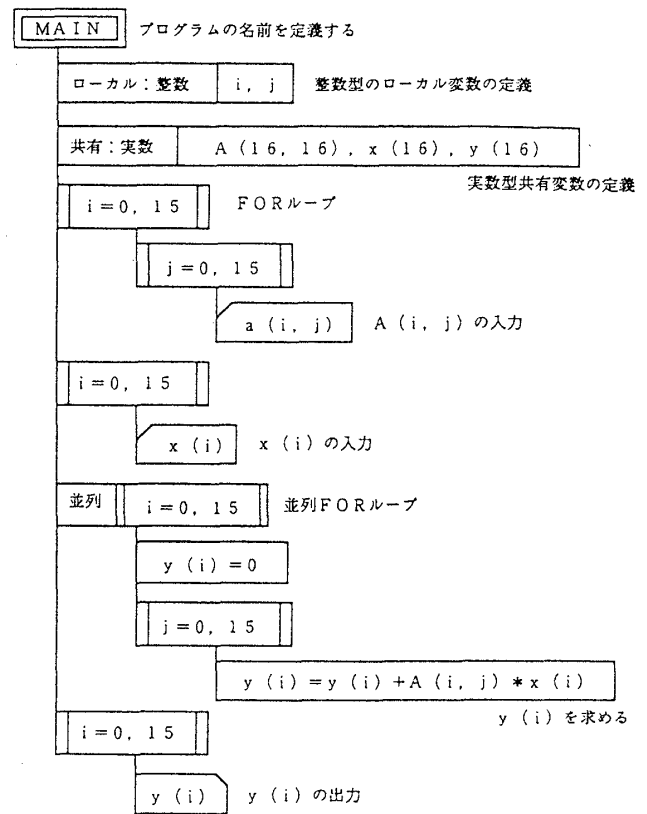


図2. ソースプログラムの例

Programming System for The Multi-Processor System 'SAKYU'

Toshimasa Suzuki, Michio Inoue, Yasuhiro Kobayashi

Tottori University

プログラム中で使用できる変数には、共有変数とローカル変数の2種類がある。共有変数とは、すべてのPUが同じように使用できる変数で、ローカル変数とは同じ変数名でもPUごとに数値の格納場所が別々になっている変数のことである。ローカル変数は、共有変数より高速に読み書きを行うことができる。並列処理を行うためには、これらの変数をうまく使い分ける必要がある。

並列処理を行わせるための方法は、FORループの並列化である。表現形式や実行結果は普通のFORループとほぼ同じだが、ループの内容がいくつものPUによって並列に処理される点が違う。

ソースリストは、テキスト形式の保存用ファイルに変換されて保存される。実行ファイルは、この保存用ファイルをもとに作成される。また、保存用ファイルはある一定の変換規則を用いることによって、C言語などの別の言語のソースプログラムに変換することも可能である。

4. 並列処理部分の変換について

並列計算機「砂丘」における並列処理は、サブP

Uによる処理ルーチンのコールという形で実行される。サブPUは、普段は処理の要求が来るのを待ち続けていて、何か要求が来ると、要求の内容に応じたパラメータを受け取り、処理ルーチンを呼び出す。処理ルーチンでの処理を終えたサブPUは、再び要求がくるのを待ち続ける。

並列FORループは、次のような処理に変換される。まずループの内容をサブルーチン化して、サブPU用の処理ルーチンとする。サブルーチンの先頭には、ループ用の変数を受け取るプログラムをつけ加えておく。次に、ループ用の変数を繰り返しの数だけサブPUに渡してからサブPUに動作要求を出す部分を作る。最後に、サブPUが全部処理を終えるまで待つループをつけて完成する。

5. おわりに

現在、基本的な部分が動作するものができていて、簡単な並列処理プログラムの作成ができるようになっている。今後は多くの利用者に使用してもらい、その意見を元にしてさらに使いやすいシステムを開発していくことにする。

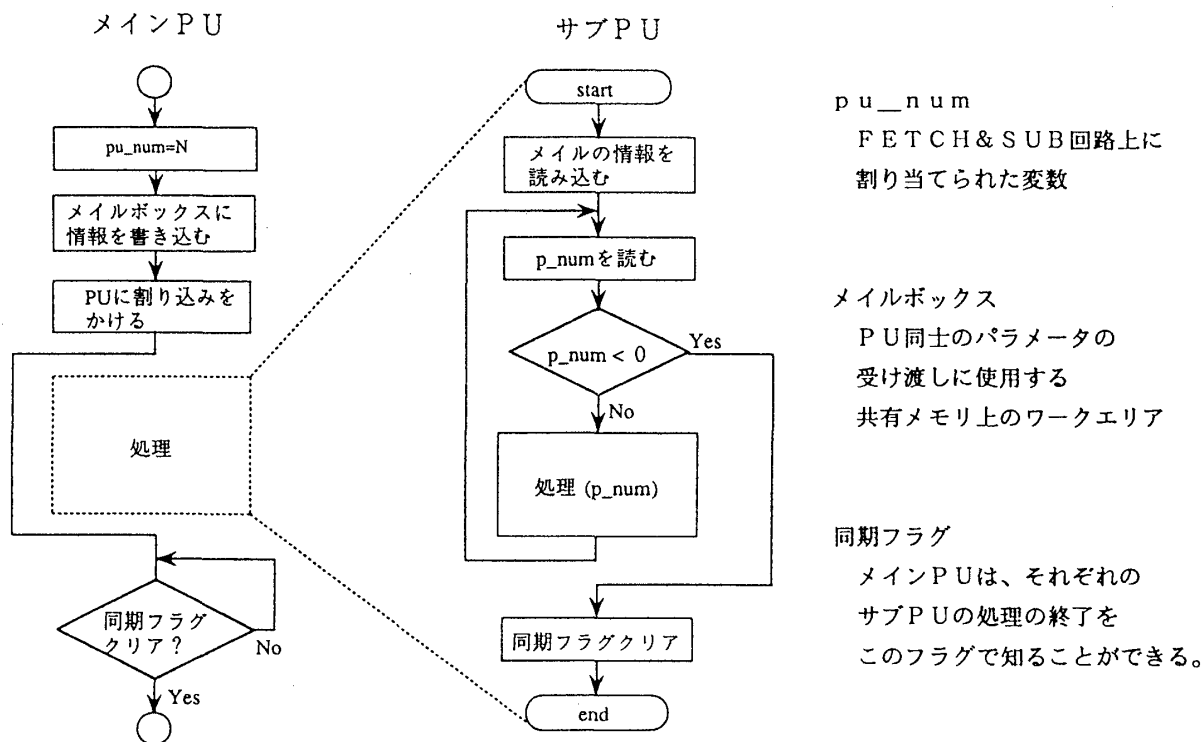


図3. 並列処理部分のフローチャート