ＰＩＭＯＳのタスク管理方式
－タスク終了時の資源解放－

松尾　正浩、　藤瀬　哲朗、　佐藤　裕幸、　近山　隆
（株）三菱総合研究所　　（財）新世代コンピュータ技術開発機構

１．はじめに

ICOTにおける、第五世代コンピュータの研究開発プロジェクトでは、並列推論マシン用の並列論理型オペレーティング・システム[1]として、PIMOS(Parallel Inference Machine Operating System)を開発中である。

PIMOS環境下におけるユーザ・プログラムの実行時には、PIMOSとユーザとが互いに協調して処理を行う必要があり、PIMOS－ユーザ間の通信を伴なう。資源管理では、PIMOSとユーザとを結ぶこの通信路の管理を行う。

本稿では、PIMOSにおける資源の管理のうちで、タスクの管理に関わる部分、タスク終了時の資源解放の機構について報告する。

２．PIMOSが扱う資源

PIMOS上では、ユーザのプログラムは、すべてタスクと呼ぶ実行単位ごとに管理する。

PIMOSとユーザ・タスクとの通信路は、すべて資源として取り扱う。その他にもPIMOSには、リダクション数の割り当て（ほぼCPU及びメモリ資源の割り当てに相当する）などの資源管理も存在するが、これらはKL1言語で管理しており、PIMOSは関与しない。

PIMOSが、資源として取り扱うPIMOS－ユーザ間の通信路は、具体的には次の様なものである。

①PIMOS本体にリクエストを送るための通信路
　ユーザ・タスクごとに必ず存在し、タスク生成時には、ユーザにとって唯一の資源である。この通信路を通して、各サブシステムへの通信路を要求する。

②サブシステムにリクエストを送るための通信路
　ファイル、ウインドウなどの各サブシステムへリクエストを送る通信路である。個々のファイルやウインドウへの通信路を要求する。

③個々のデバイスにリクエストを送るための通信路
　個々の入出力デバイスに対して、入出力要求を行うための通信路である。

３．資源の解放

ユーザが引き起こした誤りが、OSに悪影響を与えないようするためには、何等かの手段を用いて、OSを保護することが必要である。ここでは、PIMOSの保護をPIMOS－ユーザ間の通信路を監視するという観点から捉え、資源管理の問題として取り扱う。

（１）解放の必要性

通信路の両端には、必ずその通信路を監視しているプロセスが存在しており、PIMOS－ユーザ間の通信路の場合には、PIMOS領域内とユーザ領域内とに各々監視プロセスが存在する。（注：本稿では、プロセスは、特定の実体を指し示すものでなく、「実行状態にあるプログラム」という程度の意味で用いている。）

ユーザが、利用した資源を解放していない状態で、実行を強制的に終了した場合を仮定する。ユーザ領域内のプロセスは、いずれにしろ強制的に終了させられるため問題は生じないが、PIMOS側にある監視プロセスは永久にPIMOS内に残存してしまう。

また、具体的には、画面上にウインドウが残ってしまったり、クローズ処理を行っていないために、あるファイルが、再度オープンできなくなる状況が起こり得る。

このような事態は、ユーザが意図するか否かに関わらず、PIMOSに悪影響を及ぼすため、回避する必要がある。

（２）解放のタイミング

資源の利用が終了した時には、その資源を利用していたユーザは、各自の責任において、明示的に、資源の解放をPIMOSに対して宣言する必要がある。

しかしながら、資源の解放を全くユーザの責任において行うものとしてしまうと、資源を利用しているユーザ・タスクの実行中に異常が生じ、そのユーザ・タスクを強制的に終了させた場合には、資源の解放は行われない。

そこで、タスクに異常が起きて強制的に終了された場合には、必ず利用中の資源を解放する必要がある。

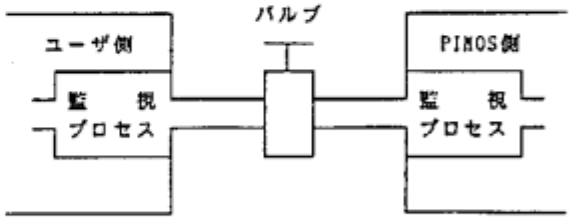
（３）解放の方法

統一的な処理手順で、すべての資源の解放を可能にするため、次のような資源解放のプロトコルを設定した。

「資源にアクセスするための通信路に対して送端より終端記号を送信する。受端では、この終端記号の検出をもって資源の解放要求とみなし、受端のプロセスによる監視を終了する。」

問題となるのは、資源を利用していたタスクが、この終端記号を送出せずに異常終了した場合である。

PIMOSでは、すべてのPIMOS－ユーザ間の通信路に、その通信路を遮断するための「バルブ」に相当する機能をその通信路の生成時に挿入している。そして、常にタスクの実行状況を監視し、タスクが異常終了した際には、自動的にこの「バルブ」を閉じて、資源を解放するような機構をタスクに持たせている。



Task Management on PIMOS
M.Matsuo, T.Fujise, H.Sato, T.Chikayama
MRI　　MRI　　ICOT　　ICOT

次に、「バルブ」の実現イメージを概説する。「バルブ」は、入出力用の通信路の他に、「バルブ」を閉じるスイッチの役割を持つ変数を用意している。

タスクを監視しているプロセスは、タスクからの終了報告を受け取ると、この変数にタスクが終了した旨のメッセージを送る。「バルブ」は、常にこの変数を監視しており、終了メッセージを検出すると、PIMOSに対して資源解放のメッセージを送り、自身は終了する。PIMOS側の監視プロセスが、このメッセージを受け取ることで、資源の解放が行われる。

例えば、「バルブ」の基本的な機能部分のコードのイメージは、以下のようになる。

```
valve(In, Out, shut) :- true | Out = {}.
valve({}, Out, _) :- true | Out = {}.
valve({Data | In-remain}, Out, Shut) :-
     Out = {Data | Out-remain},
     valve(In-remain, Out-remain, Shut).
```

注意すべき事の一つとして、この「バルブ」を挿入する位置の問題がある。「バルブ」のプロセスは、必ずPIMOS側に作らねばならない。もし「バルブ」がいずれのタスクにしろユーザ・タスク内に作られたとすれば、「バルブ」の含まれているタスク自体が強制的に終了される恐れがあり、資源の解放は保証されない。

4．資源管理の単位としてのタスク

PIMOSでは、資源の利用者として、タスクと呼ばれる論理的な単位を仮定している。これまで、タスクを一般的な意味で用い、説明を怠ってきたが、以下の記述においては、このタスクの構造について、その概略を述べる。

（1）タスクに必要な機能

ユーザ・プログラムを実行・管理するため、タスクに要求される機能のうち、資源のユーザとしての立場から要求される機能は、次の点に集約される。

①タスクは、そのタスク内で実行されたすべてのユーザ・プログラムの実行状況を常に把握し、それらの実行がすべて終了した際には、あらかじめ設定された、タスク自身の監視プロセスに対して終了報告を行う。

②タスクは、強制的に終了された場合にも自身の監視プロセスに対して異常終了報告を行う。

（2）荘園構造から見たタスク［2］

PIMOS記述言語であるKL1（核言語第一版）の言語仕様では、あるプロセス群全体の終了を検出する方法として、荘園構造を提供している。荘園を用いれば、その荘園内で実行されたプロセスすべてが終了したとき、あるいはその荘園の実行を強制的に終了させたときのいずれの場合も検出し得る。

ユーザ・プログラムの論理的な実行単位であるタスクは、実際には、この荘園構造を用いて実現している。

5．問題点と解決方法

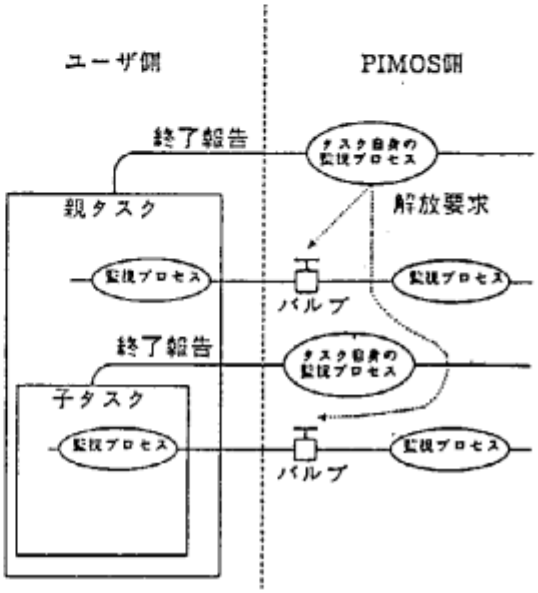
単純に「バルブ」の機構を付与するだけでは、解決されない問題点もある。それらの問題点の内のいくつかを以下に述べる。

（1）親子タスクの間での解放

ある荘園の中で、さらに荘園を生成した場合、それらの荘園同士は、親子の関係にあると考えられる。親荘園を強制的に終了させたときには、内部にある子荘園も強制終了されるが、タスクは、荘園構造を用いて実現されるため、親子タスク間にも同様の関係がある。

そこで、親タスクが終了した場合には、親タスクが利用している資源を解放するのみではなく、子タスクが利用している資源も同時に解放せねばならない。

この問題は、子タスクの資源に対して、子タスク自身の終了時に資源の解放要求を送るだけでなく、親タスクの終了時にも資源の解放要求を送れるように、親子タスクの「バルブ」間の連絡を保つことで解決している。



（2）資源の解放忘れ

資源を閉じ忘れたままユーザ・プログラムが終了した場合、ユーザ・タスク内の監視プロセスが残ってしまいユーザ・タスク自身が終了せず、デッド・ロックを引き起こすという問題がある。

この点に関しては、現在までのところ対処方法が定まっておらず、無限ループ等と同様にユーザの誤りであるとして、PIMOSは関与しない方針を取っている。

6．おわりに

現在、PIMOSの資源管理に関しては、設計、問題点の抽出に努め、その解決方法について検討を重ねている。今後、マルチPSI第２版上での実現に向けて、さらに検討を重ねていく予定である。

＜参考文献＞

［1］佐藤　他：PIMOSの概要－並列推論マシン用オペレーティング・システムの構築－
第３４回情処全国大会，2P-8，1987-3

［2］佐藤　他：並列論理型OS－PIMOS－資源管理方式－
第３５回情処全国大会，4D-3，1987-9

［3］渕　　他：Multi-PSIシステムの概要
第３２回情処全国大会，2Q-8，1986-3