

PIMOS上でのシェルの実現 ー共有資源管理ー

松尾 正浩、 藤瀬 哲朗、
(株)ニ 菱 総 合 研 究 所

佐藤 裕幸、 近山 隆
(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(2)として、PIMOS(Parallel Inference Machine Operating System)を開発中である。

シェルは、PIMOSが提供するユーザ・インタフェースの一つで、一般のユーザ・プログラムは、このシェルを用いて起動、管理するのが通常である。

本稿では、PIMOS上でのシェルの位置づけ、ユーザに提供する機能についてその概略を報告する。また、特に入出力資源の共有のための機構については、その実現方式に関しても言及する。

2. PIMOS上でのシェルの位置づけ

multi-PSI(4)上で実行されるプログラムは、大別して以下の2種類に分類される。

- ① OSプログラム(PIMOS本体)
- ② ユーザ・プログラム

OSプログラムは、ユーザが、いかなる過ちを犯してもOS自身に悪影響を及ぼさないように設計する必要がある。そして、その要求を満たすため、過剰と思われるようなOSサービスは、極力排除してある。

この立場から検討すると、シェルは、ユーザ・プログラムとして取り扱うのが妥当である。従って、シェルはPIMOS本体には組み入れられていない。

PIMOSから見たとき、シェルは、ユーザ・プログラムの実行制御をインタラクティブに行うため用意された、ユーザ・インタフェースの一つにすぎない。そのため、PIMOS上では、一般のユーザ・プログラムとして扱われる。

3. シェルが提供する機能

シェルが提供する主な機能を以下に列挙する。

3.1 ジョブ管理機能

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。一方、シェルは、シェル自身が起動したユーザ・タスクを個々のタスクごとに管理するのではなく、タスク間通信機能を利用して連結された一連のタスクごとに管理する。この管理単位をジョブと呼ぶ。

ジョブの実行環境には、フォア・グラウンドとバック・グラウンドの2種類があり、シェルが起動したすべてのジョブは、このいずれかの環境にある。(フォア・グラウンドにあるジョブのみがシェル・ウィンドウを利用できる。)

'SHELL', Implementation on PIMOS
M. Matsuo, T. Fujise, H. Sato, T. Chikayama
MRI MRI ICOT ICOT

また、ジョブの実行状態には、実行中、中断中の2種類があり、起動されたすべてのジョブは、このいずれかの状態にある。

ジョブ管理機能は、上記の実行環境および実行状態を管理するための機能を提供する。

3.2 標準入出力資源管理機能

シェルは、ユーザ・タスクを起動する際に、タスクに対して、標準入力、標準出力、標準メッセージ出力、標準インタラクションの4つの入出力資源を割り当てる。標準入出力資源管理機能は、これら4つの資源の割り当て、共有時の排他制御、アテンション処理等を司る機能である。以下に標準入出力資源の用途の概要を記す。

標準入力:

タスクが標準的な入力を行うための資源である。

標準出力:

タスクが標準的な出力を行うための資源である。

標準メッセージ出力:

タスク実行中に発生するメッセージを出力するための資源である。この入出力資源に対して送出したメッセージは、資源をアボートした際にもキャンセルされない特徴を持つ。

標準インタラクション:

タスクの実行中にユーザとのインタラクションを行うための資源であり、専属タスク中で最も近くにあるシェルのシェル・ウィンドウに接続されている。また、シェル・ウィンドウからのアテンション通知用にも使用される。

標準インタラクション以外の3つのストリームは、ユーザが起動するゴールと共に指定することが可能であるが、特に指定がなければ、起動するシェル自身が持っている標準入出力を与える(共有する)。

ただし、ユーザがゴールの中で他の入出力資源を獲得し、利用する場合には、シェルは何ら関与しない。

3.3 例外処理機能

ユーザ・プログラムの誤りによって引き起こされる種々の例外事象は、荘園の持つ例外処理機構を利用して、シェルに報告される。シェルは、これらの例外事象を検出すると、これに対処するため、シェル・ウィンドウを介してユーザに可能な対処方法を呈示し、判断を求める。そして指定された対処方法に従って、例外事象に対する処置を施す。

例外処理機能は、これら一連の機能を提供する。

3.4 コマンド・インタプリタ機能

ユーザがシェル・ウィンドウから入力した各種コマンドをシェルのコマンド文法に則り解釈し、所定の処理を行う。主な処理としては、ジョブの起動、中断、再開、強制終了、状態問い合わせ等がある。

4. 共有資源の管理

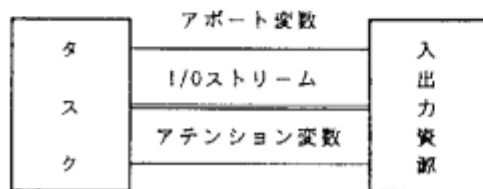
前述の標準入出力管理機能の項で述べた通り、シェルの標準入出力資源（通常はシェル・ウィンドウ）とその子供ジョブの標準入出力資源とは、共有される可能性がある。しかし、PIMOSの提供する資源管理機構には、資源の共有を支援するための機能はない。

そこで、標準入出力資源の共有を実現するため、シェルが必要な機能を提供する必要がある。

4.1 PIMOSが提供する入出力資源

現在PIMOSが提供している入出力資源は、ウィンドウおよびファイルの2種類である。

これら入出力資源は、資源を利用するタスクとPIMOSとの間の通信路として提供される。具体的には、入出力メッセージを送るためのストリーム（I/Oストリーム）、資源からアテンション発生をタスクに通知するための変数（アテンション変数）、資源に対しアクセスの中断（アポート）を指示するための変数（アポート変数）の3つを組としてユーザに提供する。[1]



4.2 共有資源の問題点

一つの入出力資源に複数のタスクからアクセスするためには、一つのストリームに対して複数のタスクからメッセージを送る必要が生じる。そのためには、ストリームにメッセージをマージするためのプロセス（マージャ）を付加することとなる。

ストリームにマージャを入れた場合、ひとたびマージャを通過したメッセージは、なんらかの工夫を加えておかない限り、マージャのどの入口から入力されたものかを識別することはできない。通常の処理においては、識別を行う必要はないが、資源をアポートする場合を想定すると、なんらかの機構を設けておく必要がある。

4.3 共有資源の実現方式

実際には、入出力資源の共有は、以下の方針の基に実現されている。

(1) ジョブ間での共有

すべての標準入出力資源は、ジョブ単位に排他制御を行う。排他制御を行うための機構は、ジョブ内に設け、シェルがそれを制御する。

(2) ジョブ内のタスク間での共有

一つのジョブ内にある複数のタスクが、同一の入出力資源を共有する場合には、シェルが自動的にマージャをそのジョブ内に挿入する。従って、一つの入出力資源とジョブ間に複数の通信路が設定されることはない。

(3) ジョブ内の一つのタスク内での共有

ジョブ内にあるタスクの一つが、一つの入出力資源に複数の通信路を設ける場合には、シェルが自動的にマージャをそのタスク内に挿入する。従って、一つの入出力資源とジョブ内のタスク間に複数の通信路が設定される

ことはない。

(4) アテンションの検出

資源からのアテンション通知の検出は、ジョブ単位に行う。アテンションを検出したジョブは、直ちにシェルに対して、アテンション発生を報告すると共に、入出力資源の使用権をシェルに返還する。これらの処理は、標準入出力フィルタと呼ばれるアテンション処理用のプロセスが行う。

(5) 標準メッセージ出力

標準メッセージ出力資源を共有する必要がある場合には、それは実際の入出力資源につながるのではなく、観ジョブ（通常はシェル）への通信路として与えられる。従って、実際の入出力資源への出力は、すべて観ジョブの責任となる。

(6) 入出力資源の獲得

ユーザは、タスク内から入出力資源を獲得する方法として、以下の2種類の方法を選択できる。

- ① 4種類の標準入出力資源（ストリーム）をすべて独立に獲得する。
- ② 標準入力、標準出力、標準インタラクションをマージした一つのストリームと標準メッセージ出力ストリームとの2つを入出力資源として獲得する。

5. 今後の予定

現在PIMOSが提供している入出力資源には、フロント・エンド・プロセッサ(FEP)[3]として利用したPSIの入出力資源管理機能に依存する部分が残存している。しかし、PIMOSの最終的なターゲットとして目指している並列型推論マシン(PIM:Parallel Inference Machine)では、PSIのような高機能の入出力機能を持つFEPを利用するか否かは、定かではなく、入出力資源の管理機構をすべてPIMOSが行う可能性もある。

その際には、シェルが行っている共有資源管理をPIMOS側に移管することも考えられる。そこで、今後は移管とした際の管理方式および実現方式について、検討を加えていく予定である。

6. おわりに

現在シェルは、開発用マシン(Symmetry)上に構築されたPIMOS開発支援システム(PDSS:PIMOS Development Supporting System)およびPSI(Personal Sequential Inference machine)上に構築された疑似処理系を用いて観意開発中である。今後は、multi-PSI上への移植を行い、'88/11月に東京で開催されるFGCS'88(Fifth Generation Computer Systems 1988)での発表に向け、さらに評価・改良を進めていく予定である。

<参考文献>

- [1] 松尾 他：PIMOSのタスク管理方式
ータスク終了時の資源解放ー
第36回情報処理学会全国大会，3D-4，1988-3
- [2] 佐藤 他：並列推論型OS-PIMOS-資源管理方式-
第35回情報処理学会全国大会，4D-3，1987-9
- [3] 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回情報処理学会全国大会，2P-8，1987-3
- [4] 瀧 他：Multi-PSIシステムの概要
第32回情報処理学会全国大会，2Q-8，1986-3