KL1による簡易OS − MicroPIMOS

中越靖行, 平野喜芳, 宮崎芳枝, 西崎慎一郎, 宮崎敏彦
(富士通ソーシアルサイエンスラボラトリ) (ICOT)

1 はじめに

我々は並列推論マシンのOSであるPIMOSを開発するためにPDSS(PIMOS Development Support System)と呼ぶKL1システムを開発した。PDSSはWAM[3]を基礎とした抽象機械語(KL1-B)[4]をエミュレートするバイトコード・インタープリタとKL1自身で記述されたユーザ・インタフェース・プログラムから成るシステムであり本稿ではこのうちユーザー・インタフェース部であるMicroPIMOSについて述べる。MicroPIMOSの大まかな特徴はKL1のみで記述されたSingleUser,SingleTaskの簡易OSであると言うことである。

2 設計方針

PDSS開発の第一の目的は並列推論マシンの開発と平行したPIMOSの開発にあり、できるだけ早くOS開発グループに提供する必要があった。このため実現する機能は必要最小限に抑え、SingleUser,SingleTaskのOSとして開発することにした。主な設計目標としては以下のものが上げられる。

1. ユーザ・タスクの失敗に対するOSの保護
2. ユーザ・タスクの実行制御
3. 良好なマンマシン・インタフェース

3 全体構成

MicroPIMOSはウィンドウ・システム、ファイル・システム、コード・マネージャ、コマンド・インタプリター、タスクと言った機能モジュールよりなるシステムである。個々の機能モジュールはシステム・ストリームと呼ばれるストリームを握っているが、このシステム・ストリームに宛先付きのメッセージを流すことにより他の機能モジュールと通信ができるようになっている。
MicroPIMOSの全体構成を図1に示す。以下では個々の機能モジュールについて概説する。

3.1 ウィンドウ・システム

MicroPIMOSのウィンドウ・システムはウィンドウへの入出力をEMACS(フルスクリーン・エディター)をフロントエンドとすることによりして実現しており、マルチ・ウィンドウもEMACSのマルチ・ウィンドウ機能を使って実現されている。
ユーザがウィンドウの入出力サービスを利用したい場合はMicroPIMOSが提供する述語を呼び出すことによりウィンドウに対するコマンド・ストリームを得ることが出来る。ユーザはこのコマンド・

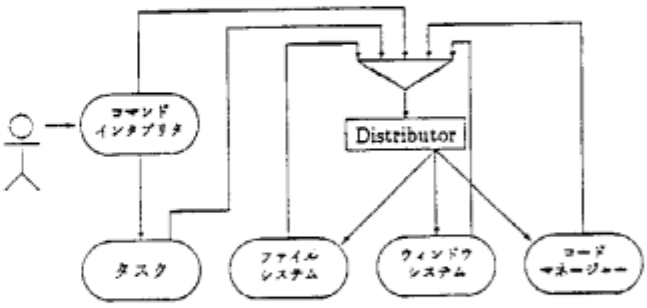

図 1: MicroPIMOS 全体構成

ストリームに種々のI/Oコマンドを流すことによってウィンドウの機能を利用することができるわけである。この時ウィンドウ・システム内にはウィンドウ管理プロセスと呼ばれるプロセスが生成され、先にユーザに渡したコマンドストリームやI/Oデバイス, Interrupt, Termination, Abortといった種々のストリームを一括管理している。ここでI/Oデバイスと言うのはフロントエンドのウィンドウと繋がるデバイスストリームのことであり、Interruptは実行の停止等、緊急なメッセージをタスクに伝えるストリームである。Terminationは資源を開放する為のフラグであり、3.5章で詳しく述べる。Abortは既にI/Oストリームへ送ってしまったメッセージを取り消すためにある。また、ウィンドウ全体の管理は一つの永久プロセスが行っており個々のウィンドウ管理プロセスをストリームによって環状に繋げてウィンドウの生成、名前による呼び出し、総数の管理と言った制御を行っている。

3.2 ファイル・システム

MicroPIMOSのファイルはUNIXのファイルを使って実現している。ファイルの入出力サービスの利用方やファイル・システムの構造はウィンドウとほとんど同じである。またこのファイル・システムはディレクトリー操作の機能も持っており UNIX のディレクトリーをユーザー・プログラム側から操作することができる。

3.3 コード・マネージャ

コード・マネージャは外部からのコマンドを解釈する部分とモジュール情報を管理するプロセスから成る。コード・マネージャの提供する機能には主に次のものがある。

1. ロードされたモジュールの名前とそのモジュール中の各種情報(例えば、他のモジュールに公開している述語名の一覧、スパ

Micro PIMOS - A tiny Operating System in KL1

Yasuyuki NAKAGOSHI [1], Kiyoshi HIRANO [1], Yoshie MIYAZAKI [1], Shin'ichirou NISHIZAKI [1], Toshihiko MIYAZAKI [2]

1: Fujitsu SSL, 2: ICOT

イ・モードになっている述語名の一覧) を要求に応じて答える機能。

2. コマンド・インタプリタからセーブ・コマンドを使ってセーブしたモジュールのオート・ロード機能。

コード・マネージャを通してモジュールをロードすると、そのモジュールの情報 (例えば、他のモジュールに公開している述語名の一覧、このモジュールが既にセーブされているかどうかなど) を内部状態として持つプロセスが生成され、モジュール管理プロセスが持つテーブルにストリームで繋がれる。これ以降このモジュールに対するセーブやロードまたはモジュール中の各種情報の請求といった要求はそのプロセスが自分の持つモジュール情報に応じて処理する。

3.4 コマンド・インタプリタ

コマンド・インタプリタはユーザに対し MicroPIMOS の他のモジュールの機能を種々のコマンドを通じて提供する会話型の実行環境である。コマンド・インタプリタの提供する主なコマンドはモジュールのロード・セーブ及びトレースやスパイモードの設定などである。また

ModuleName:Goal

と入力するとタスクを生成し、そのタスク内で ModuleName で示されるモジュール中の Goal を実行する。この他 MicroPIMOS には実行中のタスクのリダクション数を聞く機能とタスクの実行を中止させる機能がある。

3.5 タスク

MicroPIMOS におけるタスクは KL1 が持つ荘園 [2] の機能を中心に実現される。タスクの構成を図 2 に示す。
タスクは荘園とその荘園を監視及びコントロールするタスクモニタから成る。タスクモニタと荘園は二本のストリームで繋がっており、タスクモニタから荘園へ向かうストリームをレポートストリームと呼び、荘園からタスクモニタへ向かうストリームをコントロールストリームと呼ぶ。レポートストリームは荘園内で起きた例外の情報などが上がってくるストリームで、ユーザプログラムからの OS との通信要求などもここから上がってくる。一方コントロール・ストリームは荘園をコントロールするためのストリームであり、ここに start, stop, abort, statistics(N) 等のコマンドを流すことで荘園内のゴールの実行の中止や実行途中のリダクション数の参照などを行える。タスクモニタの主な仕事を以下に示す。

1. インタラプト処理
   コントロール・ストリームによる荘園操作により、タスクの実行中止や実行途中のリダクション数の参照といったインタラプト要求に応える。

2. 例外処理
   タスクモニタが受け取る例外にはインテジャ・オーバーフローなどの演算異常や OS との通信要求があり、その種類に応じて例外情報の表示やタスクの強制終了あるいは、他の OS のモジュールへの要求の転送などを行う。要求の転送にあたってはその要求にタスクモニタが管理しているフラグ (タスクの終了の通知用) を付加して転送する。またこの時、図 2 の様にコマンドモニタと呼ぶプロセスを荘園内に生成させる。このプロセスはユーザからのコマンドのプロトコルをチェックし、おかしなコマンドが OS に流れ出さない様にしている。この他の例外処理としてはコードのオートロード機能の実現がある。コマンド・インタプリタから save(ModuleName) や save_all コマンドを使ってセーブしたモジュールはコード・マネージャによって所定の場所にセーブファイルが作られ、そのモジュール名で管理される。タスクモニタはこれを利用し、自分が管理している荘園内で モジュールが存在しないと言う例外が発生した場合にはコード・マネージャに対してこのモジュールをロードするようメッセージを送ることによりオートロード機能が実現されている。

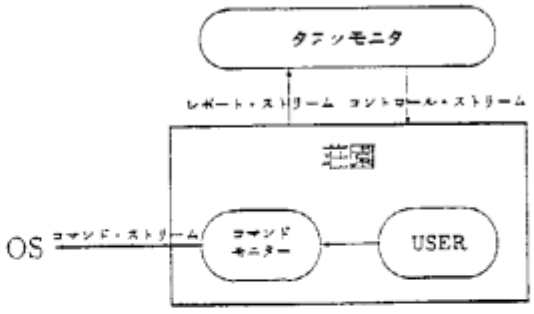

図 2: タスクの内部構成

3. 資源の開放
   MicroPIMOS ではタスクの終了に合わせて各プロセスが管理していた資源を自動的に開放する様にしている。各資源はそれを管理しているプロセスへのストリームを閉じることにより開放される様になっているが異常終了などによりそのストリームを閉じることが出来なかった場合でも自動的に開放される。これはタスクが終了するとタスクモニターがウィンドウやファイルを管理しているプロセスに前出の終了フラグを用いてこのことを通知し、使用していた資源を開放させるためである。

謝辞

日頃御指導頂いている ICOT 第 4 研究室の方々に感謝します。またユーザとして貴重なコメントくださった木村康則、関田大吾、越村三幸、吉田かおる の各氏に感謝します。なお本研究は第五世代プロジェクトの一環として行われました。

参考文献

[1] 平野他: 汎用計算機上の KL1 処理系におけるメモリ管理とデッドロックの検出, 本大会予稿集 7H-5

[2] 佐藤他: PIMOS の概要 –並列論理マシン用オペレーティング・システムの構築–, 情報処理学会第 34 回全国大会,2P-8,1987-3

[3] D.H.D.Warren: An Abstract Prolog Instruction Set. Technical Note 309 SRI International

[4] 木村他: 並列推論マシン PIM –KL1 の抽象命令仕様とコンパイラ–, 情報処理学会第 34 回全国大会,2P-1,1986-3

[5] PDSS グループ: PDSS- 言語仕様と使用手引き,ICOT TM-437,1988