

Common LISP仕様に基づく Concurrent LISP処理系の実現

2P-3

藤田岳久 阪口哲男 杉本重雄 田畑孝一

図書館情報大学

1. はじめに

Common LISP仕様[1]に基づくConcurrent LISP(以下CCLと略す)は、我々の設計した協同処理記述用言語である[2]。本稿では現在開発を進めているCCL処理系について述べる。

2. Common LISP仕様に基づくConcurrent LISP

CCLは逐次処理に関してCommon LISPの仕様を満たし、かつ従来のLISPに基づき開発した協同処理記述用言語Concurrent LISP[3]のプロセス定義に準拠した仕様を持つ協同処理記述用言語である。

CCLではプロセスを

「プロセスはそれだけで完備した能力で形式(form)を評価する実体である。」

と定義する。プロセスは動的に起動され、親子関係を持ち、共有変数やメッセージ送受機能を用いたプロセス間コミュニケーションを行う。協同処理機能に関しては基本並行処理記述用スペシャル・フォームを3つ導入する。また、本処理系ではマルチウインドウ機能を活用した良好なユーザインタフェースを提供する入出力機能等を備える。

3. 実現環境

実現環境は以下の通りである。

ハードウェア: ソニーNEWS

オペレーティングシステム: UNIX 4.2BSD

マルチウインドウシステム: X-Windowシステム

使用言語: C言語

処理系実現は、CPU依存性をなくし移植性を上げるという点を考慮しC言語のみで行う。またマルチウインドウシステムに関しては、機種依存性の低いX-Windowシステムに基づくこととした。

4. 処理系の構造

4.1 概要

CCL処理系は図1に示す様に、CCL仮想マシンとその上のCCLインタプリタから成っており、階層構造を持つ。CCL仮想マシンは基本データ構造や領域、及びそれを操作する諸機能を提供し、それらを利用してCCLインタプリタは種々の関数機能を実現する。また、入出力操作に関する並行性を高めるため、CCL処理系はX-WindowシステムとCCL本体プロセスとのインタフェースであるウインドウ制御プロセスを持つ。

4.2 CCL仮想マシン

CCL仮想マシンはCCL処理系の中で扱うデータ構造(表1参照)と制御用の関数、及びCCL処理系に対応した入出力関数を提供する。

CCL仮想マシンの構成要素は以下の通りである。

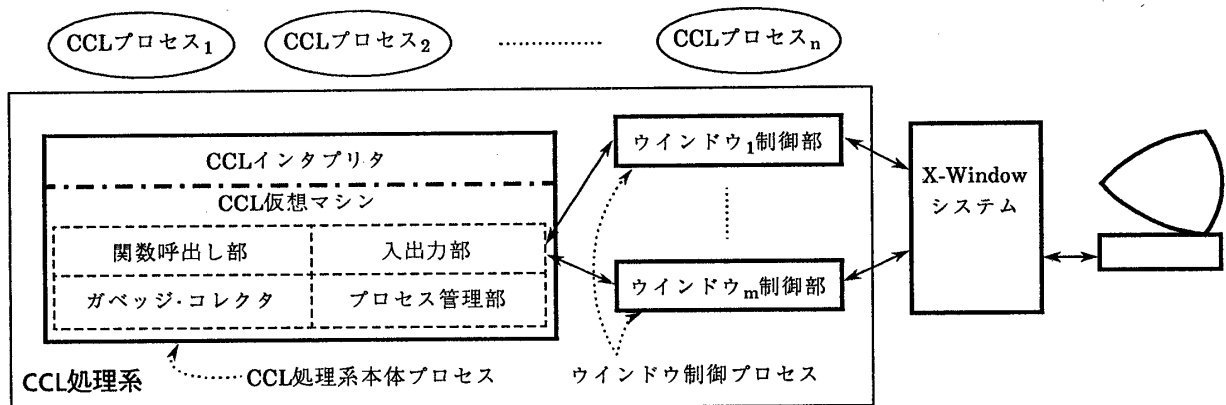


図1 CCL処理系の構造

Development of Common LISP-based Concurrent LISP

Takehisa FUJITA, Tetsuo SAKAGUCHI, Shigeo SUGIMOTO, Koichi TABATA

University of Library and Information Science

表1 CCL仮想マシンの提供する主なデータ構造

| データ構造 | 機能 |
|------------------------|--------------------------------------------------|
| プロセス・コントロール・ブロック (PCB) | 各プロセスの実行時の情報や親子リンク、プロセス間コミュニケーションのためのメールアドレスを持つ。 |
| ランタイム・スタック (RTS) | 関数実行時の環境を保持する。リンクドスタックの形で実現する。 |
| セル | リスト・数値など様々なデータを表す。 |
| 大域変数表 | 各プロセスの大域変数を持つ。 |

- ・プロセス管理部
- ・関数呼出し部
- ・入出力部
- ・ガベッジ・コレクタ

● プロセス管理部

並行処理を行うための機構で、プロセスの生成・終了、プロセスの実行、プロセス切替えを行う。以下の3つの部分から成る。

・プロセス生成管理部

プロセス生成の際、プロセス管理部が制御を行うために必要な情報をプロセス毎に保持するプロセス・コントロール・ブロック(以下PCBと略す)を確保して初期化し、プロセス管理のためのPCBチェーンに加える。

・プロセス切替え管理部

プロセスの切替え、待ち状態プロセスの待ち条件判定、プロセス間の相互排他制御を行う。

プロセス切替えはプロセス内で一定回数以上の関数呼出しがあった時、プロセス終了時、及びプロセスが入出力待ちになった時行う。CCLではどの実行可能プロセスも同じ優先度で実行し、プロセス切替えには最も単純なラウンド・ロビン方式を採用する(図2参照)。

CCLは相互排他制御を行うために、他のプロセスに対し排他的に実行権を得るためのスペシャル・フォームを持つ。プロセスがこれを実行している間はプロセス切替えを行わない。

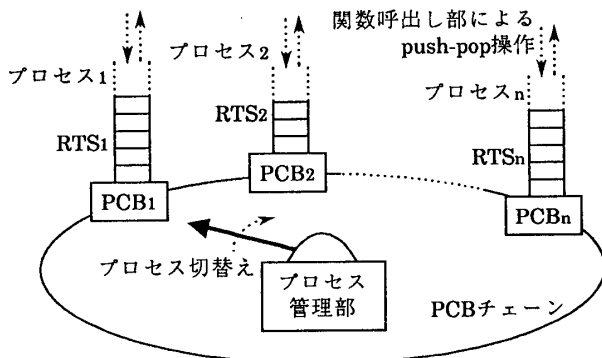


図2 プロセス管理部とPCB、RTS

・プロセス終了部

プロセスの終了処理を行う。終了状態フラグをセットし、実行中の子プロセスを持っている場合はそれらを強制終了する。終了したプロセスのPCBは主プロセス終了までPCBチェーン内に保存し、他プロセスはその情報を得ることができる。

● 関数呼出し部

関数の呼出しや関数からのリターンに必要な処理を行う。各プロセスはそれぞれ関数の呼出し状況を記録しているランタイム・スタック (RTS) という実行時スタックを持つ。関数呼出し部はこのスタックを用いて関数呼出しの機能を実現する。

● 入出力部

ファイル入出力、ウインドウ制御プロセス群への入出力などを行う部分である。UNIXが提供している入出力機能に加えて、プロセス管理部へのプロセス切替え要求を行うために必要な機能を持つ。

● ガベッジ・コレクタ

CCL仮想マシンはCCLインタプリタの要求に応じてセルを供給する。使われなくなったセルはガベッジ・コレクタによって回収され再利用される。

4.3 CCLインタプリタ

Common LISPの仕様を満たす関数群より構成される。CCL仮想マシンが提供するデータ構造や制御機能を用いてCCLプログラムの解釈実行を行う。

4.4 ウインドウ制御プロセス

CCLでは、文字入出力を行うためのウインドウを準備する。これを実現するために、CCL処理系はUNIX上に複数のウインドウ制御プロセスを持つ。これは、ウインドウに対する制御を単純化しファイルと同様に扱えるようにするために、ウインドウを直接制御する部分を本体より分離してそれぞれのウインドウにひとつずつ制御部を対応させたものである。ウインドウ制御プロセスとCCL本体プロセスとのコミュニケーションにはバークレイ版UNIXにおけるプロセス間通信機構を用いる。

5. おわりに

現在までにCCL処理系の基本機能の実現を終了しており、種々の機能の実現を進めている。

参考文献

[1] Steele, Guy. Common LISP: The Language. Digital Press, 1984, 465p.
 [2] 阪口哲男, 杉本重雄, 田畑孝一. "Concurrent Common LISP". 情報処理学会第37回全国大会講演論文集(I). 情報処理学会編. 1988, p.648-649.
 [3] Tabata, K.; Sugimoto, S.; Ohno, Y. Concurrent LISP and Its Interpreter. JIP. Vol.4, No.4, p.195-202(1982)