

A I P - L I S P : (4) 入出力システム

7P-4

河辺和作 中村明
(株)東芝0. はじめに

Common Lispの入出力は、豊富な機能を持つストリーム機構をベースにした体系を持っている。

我々は、AIP-Lisp上に、ほぼ、Common Lispの全仕様をカバーする入出力機能を実装した。今回は、その概要について報告する。

1. 実装の概要

AIP-LispのターゲットであるAIPは、独立した計算機システムではなく、ホスト計算機に付加されて動作するアタッチドプロセッサである。

ユーザプログラムレベルの入出力インタフェースとしては、UNIXと外部仕様の上で互換性があるC言語用の標準ライブラリ関数が用意されているので、ユーザは(オーバーヘッドを別にすれば)AIPの接続形態を特に意識しなくても良いようになっている。

AIP-Lispの入出力関数の実装にあたり、作成の容易さ、保守性を考慮して、最も低レベルな部分(1文字単位の入出力等)では、この標準ライブラリ(fgetc, fputc等)を利用している。

また、delete-file等、本来、OSコマンドを実行することによって実現可能な関数の実装は、ホスト計算機にOSコマンドを送りこんで実行させる方式を採用し、作成の負担を軽減した。

上位の入出力関数は、すべてLispで記述されており、拡張・変更が容易に行えるよう配慮されている。

2. 拡張機能

AIP-Lispでは、通常の入出力関数以外に、ホスト計算機上でプログラム(コマンド)を実行し、結果を(パイプ)ストリームを介して、逐次、受け取る関数(open/closeのインタフェース)が用意されている。

これは、AIP-Lisp内部でも、"directory"をはじめとする、いくつかの実行時ルーチンを実装するために使われているが、この機構は、ホスト計算機とAIPを有機的に組み合わせたプログラムを記述する上で非常に有効であると考えられる。

3. 今後の課題

現在、AIP-Lispの入出力は、機能的には、Common Lispの全仕様をほぼ実現しているものの、動作速度は決して充分なものではない。AIP-Lispを用いてプログラム開発を行う場合、入出力の性能は、作

業能率に無視できない影響がある。

現在のAIP-Lispでは、低レベルの入出力操作は全て、C言語で記述された標準ライブラリによって行われ、Lispレベルでのバッファリング操作は行われていないが、入出力の速度を改善するため、見直すことを検討している。

4. まとめ

以上、AIP-Lispの入出力機能について、実装の概要及び今後の改良方針を報告した。

今後に予定されている、プログラム開発環境の整備に合わせて、作業能率に影響の大きい入出力機能の性能向上を図っていきたい。

参考文献

- [1] 齊藤 他: "AIワークステーションの開発思想" 第1回人工知能学会全国大会(1987)
- [2] 戸辺 他: "AIP-ホスト間のインタフェース" 情報処理学会第37回全国大会(1988)
- [3] G.L.Steele Jr., "Common Lisp the Language", Digital Press, 1984