

AIP-LISP: (5) C言語インターフェイス

7P-5

佐藤祐治° 星野康夫°° (株)東芝ソフトウェアエンジニアリング °° (株)東芝

0. はじめに

AIP-LispにC言語とのインターフェイス機能を開発し、AIP-Lisp第1版でサポートされないI/O機能や、AIP上の処理プログラムの実行効率を向上させる手段を提供した。

本報告では、AIP-LispのC言語インターフェイスの概要とその実現方法について述べる。

1. 概要

AIP-Lispでは、スペシャルフォームextfuncallを用意して、C言語で記述された関数(これを外部関数と呼ぶ)とのC言語インターフェイス機能を提供している。

ただしC言語インターフェイスは、Lisp→C言語の一方に限定される。

AIP-Lisp環境におけるC言語インターフェイスの位置づけを図1に示す。

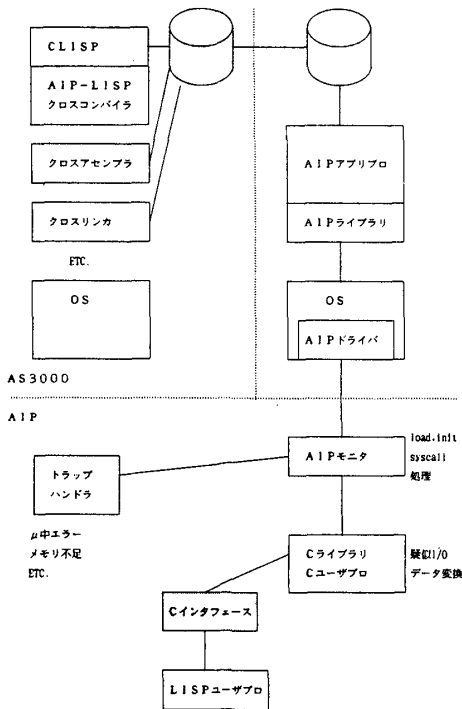


図1 AIP-Lisp第1版環境

2. 実現

2.1 extfuncallの実現

AIP-Lispでは、外部関数の呼び出しを実現するため、つぎのスペシャルフォームをサポートする。これは、Common Lisp仕様に対する機能拡張である。

```
extfuncall entry-routine return-type &rest (arg)*
```

extfuncallの呼び出しは、コンパイル時にコンパイラによってas:extfuncall\$の呼び出しに展開される。コンパイラは、extfuncallの第1引数(外部関数の名称)を対応するラベルを表現するスペシャルフォームに変換し、第2引数(外部関数の返す値の型シンボル)を、対応する数値に置き換える。

Common Lispの言語仕様では、applyの第1引数としてスペシャルフォームを渡すことが許されないので、外部関数を呼び出すとき、引数の数が実行時に決まるような場合は、ユーザプログラムの中で、as:extfuncall\$の引数を陽に記述しなければならない。

例:

(extfuncall "foo" 'fixnum x y) は、次のように展開される。

```
(as:extfuncall$
 (user::@ "foo")      ;; fooのエントリーポイント
 4                    ;; リターン値の型
 x                    ;; 外部関数の第1引数
 y                    ;; 外部関数の第2引数
```

2.2 AIP-Lisp組み込み関数の実現

AIP-Lispの一部の数値演算関数や低レベル入力関数はC言語で記述されている。これらもextfuncallによってAIP-Lisp組み込み関数として提供されている。

3. 性能評価

extfuncallによる外部関数呼び出しの性能評価を、C言語側にある関数をLisp側からAIP-Lispの外部関数として呼び出す場合と、C言語で直接呼び出す場合の処理時間の比較により行った。これよりextfuncallの時間的なオーバーヘッドは、非常に微小なものであった。

4. おわりに

AIP-LispのC言語インターフェイス機能の開発にあたり、その整合性を重視し、実行時のオーバーヘッドを極力抑える様に設計した。またAIPにおけるLispプログラムのデータアクセスをC言語でも行なえる機能を提供したことにより、AIPアプリケーションプログラムの記述が容易なものとなっている。