

2D-6

プロセス間通信として TCP/IP ストリームを用いた Lisp - C プログラムの融合

織田 一晃

山本 強

北海道大学工学部 北海道大学大型計算機センター

1 はじめに

GUIなどのライブラリの多くはC言語で書かれ、lisp 処理系からの利用は出来ないことが多い。それらのソースコードを lisp 言語に書き直すのは大変な労力を要する。

lisp 処理系や C 言語のライブラリに動作機種の制限があって、これらの組み合わせを単一機種上で実現出来ない場合もある。

ここでは、プロセス間通信として TCP/IP ストリームを用いて C 言語のプロセスと lisp のプロセスを結びつけ、lisp からの間接的な C 言語関数呼出しを実現することによりこれらの問題の解決を試みた。

なお、lisp 処理系には HCl (Hokkaido Common Lisp) を使用した。

2 HClからのC言語関数の呼出し

HCl からの C 言語関数の呼び出しは、図1のような仕組みになっている。

HCl は、TCP/IP ストリームを標準で提供している。HCl 上で C 言語関数の呼出し要求が生じると、TCP/IP ストリームを用いてそれを C 言語で書かれたライブラリドライバに伝え、ドライバ側で実際の呼出しが実行される。関数一つ毎にそれを処理する為のルーチンが必要である。

ストリーム上では、呼び出す関数とその引数に関する情報が交換される。この際、関数名をそ

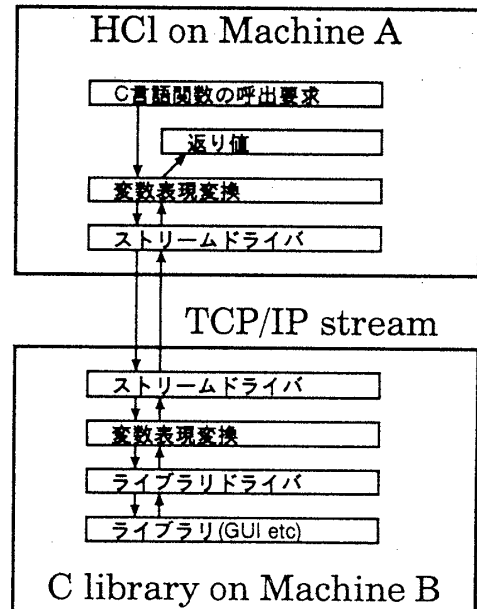


図 1: 関数呼出しの仕組み

のままストリームを通じて渡しても C 言語上には lisp での eval のような評価機能がないので、直接関数呼出しには使えない。関数名に対応する id 番号のテーブルをつくり、その id にしたがって関数呼出しをするようにした。

3 引数と変数の処理

HCl と C 言語では、変数の内部表現が違っているので、内部表現をそのまま交換はできない。たとえば、整数では HCl の fixnum は 24bit 表現であり、C 言語側の 32bit 変数を表すには bignum を使う必要が生じる場合があるが、bignum は 32bit よりもおおきな数を扱えるので over flow に関する処理が必要になる。また、小数でも HCl に実装

されているのは double のみである。

構造体や配列などの場合は C 言語と HCl とではメンバーの表現方法自体が違うので、値を渡すときにはそれぞれのメンバーごとに代入を行う必要がある。

陽にポインタ表現をつかう必要が生じた場合は、C 言語側のプロセス上に変数領域を確保して、そのポインタ値を使う。このポインタ値は HCl 上では整数として扱う。ポインタの指し示す値を HCl 上で使うときには、ポインタ値をもとにライブラリドライバからその値をコピーしてくる。

4 マクロ定義の処理

C 言語のマクロにおける define 置換文はコンパイルの前段階でのプリプロセッサによってのみ展開される。よってライブラリドライバ内で実行時にマクロ定義されている文字列が HCl 側から流れてきてもそれを正しく変換して関数呼出しをすることができない。例えば、C 言語用のソースファイルに

```
#define BLACK 0
color(BLACK)
```

とある場合、BLACK はプリプロセッサによって 0 に置き換えられ、実行時には BLACK というラベル名には意味がない。よって実行時に HCl 側から BLACK という情報を送っても意味がない。

HCl のプログラム内でこれらのものに相当する定義を適宜しておき、ライブラリドライバに渡す引数はすべてマクロ展開されたあとのものに相当するようにしておく必要がある。

5 実際例

SGI の IRIS 上に実現されている GL と呼ばれる Graphics Library は、3 次元ベクトルやポリゴンの投影など、各種グラフィック処理を簡単か

つ高速に実現することができるが、HCl の IRIS への移植はなされていない。

HCl は、MC680x0 をモデルとした仮想マシン上のアセンブラで書かれているため、MC68040 を搭載した NeXT Station Turbo では高速に動作する。

このような状況下で、NeXT 上で HCl を起動し、IRIS 上で GL の関数の一部を呼び出すプログラムを起動し、IRIS のコンソール上に GL を用いた画面を生成した。

IRIS に用意されている C 言語は ANSI 規格のため、システムで用意されているヘッダファイル内に関数引数のプロトタイプ宣言がなされているので、その変数型情報をもとに変数表現変換部を作成した。

しかし、event による関数の call back に関しての考慮がなされていないため、効率的な event 処理ができないという問題点が生じた。

6 おわりに

lisp から C 言語ライブラリを呼び出す方法について述べた。ライブラリを lisp 上に直接移植するよりも簡単にライブラリを利用することができる。TCP/IP ストリームを用いることによって HCl が動かない機種上のライブラリも利用できる。

今後の課題として、関数の call back に関して考慮する必要がある。また、ヘッダファイルをもとに、関数呼出しを処理するプログラムを自動的に作成する機構が必要となる。

参考文献

- [1] B.W. カーニハン / D.M. リッチー著 石田晴久訳 プログラミング言語 C 第 2 版
- [2] Graphics Library Programming Guide
Silicon Graphics Inc.