

資料： パネル討論会「記号処理マシンは生き残れるか？」

パネラー： 井田昌之（青学大）、小方一郎（電総研）、久門耕一（富士通研）
中島浩（三菱電機）、日比野靖（NTT）、丸山勉（日本電気）

今回の討論会においては、討論に先だって、各パネラーの立場を明確にいただいた。特に、「記号処理マシンは生き残れるか？」という問に対する回答を各パネラーに示していただき、それらを本討論会の基礎とすることにした。以下は、各パネラーから寄せられた回答をまとめたものである。

この文章と同じ物は、討論会の10日ほど前に、あらかじめ各パネラーにお送りしてある。したがって、本討論会では相手の論点を踏まえた上での議論が展開されるであろう。また、自分の議論の弱点を補強したり相手の議論の欠点を突いたりして、さらに白熱した討論も期待できる。

井田

（前置き：記号処理マシンという用語ではなく自分が慣れているLisp及びLispマシンという用語を用いる。）

（1）Lispマシンは生き延びるか？という設問は

1. 研究対象として生き延びるか？というテーマと
2. 産業として生き延びるか？というテーマの2つに分けたい。

1. についてはLispマシン関連で研究・試作を要するテーマはあり、まだやることはたくさん残っている、2. についてはLispそのものに対する意識の問題に関連する、というのが私の直接の答えである。ここでは2. について考えを述べる。

（2）一般に、いくつかの考え方・信念がある。それらをいくつかあげると、

- i. Lispですべてできる。そして自分のコンピュータソフトウェアはすべてLispで

あるべきである。

- ii. Lispで開発したい（あるいは研究したい）。しかし、できた後はCで書き換えられてしまうことには目をつぶろう。
- iii. Lispで開発して、各々に都合のいい他言語で書き換えられていい。
- iv. Lispで書かれたソフトがある。それを使えればいい。
- v. 趣味でLispを使っている。生業は別。
- vi. 言語にはこだわらない。自分のやりたいことをケースバイケースで良いと思うものを使って利用する・開発する。Lispはその中に入っている。

（3）Lispマシンの利用形態

1. 単独単機能（専用機）
2. 単独汎用（スタンドアローンの作業台）
3. 単独開発試作and デリバリ（開発には向いている）
4. ネット汎用
5. ネット専用

汎用の夢は不成立（？）：すべてのことを1つのパラダイムでなそうというのはダメ（？）

（4）ソフトウェアライフサイクルへの見方

- A. 開発then使用：「プロダクト」の世界
- B. 「インクリメンタル」な世界

（5）UI構築ツール・環境は、Lispに向けたテーマである。

その背景：

1. UI開発は試行錯誤になる。
2. 定義環境と実行環境の連続性が鍵になる。
3. 単に、視覚的な配置、メニュー的な機能選択では済まない。手続きのな付加がほしくなる。
4. 会話型の開発がいい。
5. （UI開発者ではなく）利用者が更にカスタマイズしたくなる。

産業としての共通基盤：

1. 機械アーキテクチャ
2. OS
3. 言語(?)
4. ウィンドウシステム / UI

Lispマシンの場合、ウィンドウツールキットの高速性を達し易い。

(6) 私自身は現在(3)の5.の研究を進めている。すなわち、Lispでアプリケーションを書くと、それが、そのままネットの中で生きる、という仕組みである。YYはそれについてのプロジェクトである。

YYonXでは、「自分のX端末で隣の部屋のUNIXにあるたぐさんの道具を使いながら、それらの中にLispマシンの上で動くソフトを同時に利用する。」ようなことができる。

「No Migration but Co-existence」

小方

生き残れない：

やっぱり RISC (VLIW or SuperScaler) が面白いのじゃないか。逆にこのようなアーキテクチャを利用したコンパイラの開発が、専用マシンの開発より面白くなりそうに思う。

24 bit pointer + 8 bit tag なら、専用命令は特にはいらぬ。UTILISP が、そうでしょ。32 bit pointer + 8bit tag なら、TAG 命令付きのアーキテクチャもいいみたい。SPUR がそう。いっそのこと 48 bit pointer + 16 bit tag だったら、またまた専用命令はいらなくなるのではないか？いずれにせよ、SPUR はいい線いっているのじゃないか。

ただ、現在の RISC は C 言語 + UNIX 専用マシンの性格が強いようにも思う。RISC はより「バランスの取れた」アーキテクチャに進化していこう。

専用マシンは、汎用 VLSI CPU と標準 OS には立ち向えないと思う。これらは、これからますます莫大な開発努力がかさねられ、性能が上がっていくであろう。VLIW とか、pipeline とかの技術も採り入れられて来ている。

そういう意味で、ますます差は開いていくように思う。

もう1つの話題は、専用マシンの「特殊な機能」は有用か？これに対する答えは、1. プログラムの静的解析 2. 適切なキャッシュの使用で「平均的なプログラム」は「特殊機能」がなくても速くなるはず。ただ、例えば、「一様な応答性」(ユーザインタフェースの観点から)とかからすると、「性能のパラッキがない」というのも1つの設計基準であるべきかもしれない。

死に絶えるべきだ：

これは、商品としては市場が決めること。(もちろん、研究することに関しては誰も決める必要はない。)

久門

記号処理マシンは生き残れるか？

(並列計算機として)

並列計算機が数値計算機だけになってしまうのは困るが、数値計算が出来ない並列計算機は、生き残れないだろう。

大規模な記号処理の問題の多くは、最適化問題を含んだものが多く、そのための計算(多くは整数の加減算と乗算。浮動小数は必要ないかも知れないけれど)を計算する必要があるから。

(個人的には、回路設計やLSIレイアウトのCADなんかは要求も有って数値計算っぽくなくて良いと思うけれど)

中島

命題1：記号処理マシンは生き残れるか？

回答1：生き残りたい。

命題2：記号処理言語は生き残れるか？

回答2：生き残って欲しい。

理由2：私にとって最も書きやすい言語だから。

命題3：記号処理マシンの生き残り戦略は？

回答3-1: 敵(汎用マシン)のふところに飛び込む。具体的には;

- (a) Attachment に徹する。
- (b) 「Cだって速い」ようにする(手を抜かない)。

回答3-2: 敵が少ない場所で戦う。具体的には、記号処理言語は並列処理に向いているという信念に基づき、並列マシンに全力を傾ける。

日比野

記号処理マシンは“生き延び”、将来、大発展する(ことを希望します。):

現在市場にある“記号処理マシン”は、新しい“種”の誕生過程にあるものと考えられます。したがって、滅びるものもあるでしょう。“人類”も“小さな哺乳類”の誕生から進化したもので、爬虫類全盛の時代には、マイナーな小動物に過ぎなかったのです。

このように考えるのは、現在のマシン(RISC, CISCに限らず)、およびプログラム言語は“抽象度”が低すぎるところで処理を行っていると見るからです。その意味では、“C”が諸悪の根源です。“C”の氾濫で時代は20年逆もどりしました。

このような“抽象度”の低いマシンおよびプログラム言語で、将来期待される高度のプログラミングに耐えられるのでしょうか。

LISPには大きな可能性がありましたが、Commonlispの出現で進歩した面と退歩した面があると思います。

私の“理想”は、“LISP”プラス“高いレベルの抽象性の導入”により、マシンのアーキテクチャを意識しないでプログラミングを可能にすることです。プログラム言語の“意味(解釈)”は、“言語の世界の概念”のみで“簡潔に”説明できることが必要です。

このためには、マシンのアーキテクチャを、より“抽象度”の高いレベルに設定するべきであると考えます。

技術的な可能性については、討論時に述べ

ます。

丸山

「記号処理マシンは生き残れるか?」

言語マシンとしての記号処理マシンは汎用プロセッサに吸収されると思う。以下にその理由を述べる。

従来、記号処理マシンは、

- (1) タグ分岐
- (2) 専用命令
- (3) マイクロ命令による命令レベル並列処理
- (4) その他 高速なメモリ等のハードウェアサポート

によって、汎用マシンに対してより高速な処理を実現してきた。しかし、(1)による速度向上は高々30%程度であり、(2)は処理速度にはほとんど寄与しない(プログラミングの容易さには貢献大)。

従って、(3)、(4)によって高速な処理を実現してきたことになる。しかし、(4)については、近年の汎用マイクロプロセッサは高速なキャッシュメモリ等を装備しており、このようなハードウェアによる優位性は既に失われている。また、(3)についても、汎用マイクロプロセッサに命令レベル並列処理が導入されつつあり、次世代のマイクロプロセッサにおいては命令レベル並列処理が主流となると予想される。

このため、記号処理マシンの汎用マイクロプロセッサに対する優位性はタグ分岐のみとなる。命令レベル並列処理を行なった場合のタグ分岐による速度向上は、約40~50%となると考えられるが、速度向上の観点からは、実際には2ビット程度を用いた分岐命令があれば十分である。

従って、汎用マイクロプロセッサに2ビット程度を用いた分岐命令が導入されることはあり得るが、記号処理マシンが、専用マシンとして今後生き残る可能性はないと考えられる。