6D－5

Common Lisp処理系における動的記憶管理の視覚的モニタリング

五月女 孝　　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

Common Lisp処理系における動的な記憶領域管理の振舞いは、一般に把握するのが困難である。この記憶領域管理を視覚的にモニタリングすることは、Lispプログラム実行中のConsセルの消費やガーベジコレクション(GC)の振舞いを把握できるだけでなく、プログラムのデバッグ作業などにも役立つものとなる。

本研究では、HCl (Hokkaido Common Lisp)を用いて、Common Lisp処理系における記憶領域管理の視覚的動作モニタの製作を試みた。

2　HClの記憶領域管理

HClは北海道大学で開発された小型のCommon Lisp処理系で、次の方針に沿って設計された。

1. 実行時の必要記憶領域を小さく抑える
2. コンパイル後の実行速度は他の手続き型言語と同等
3. インタプリタの速度は従来の非Common Lisp系の処理系と同等

Visual Monitoring of Dynamic Storage Management in Common Lisp
Takashi Saotome, Tsuyoshi Yamamoto
Hokkaido University

これらの方針の中、HClでは実行時の記憶領域を小さく抑えるために、単一ヒープ2領域記憶管理法を用いている(図1)。

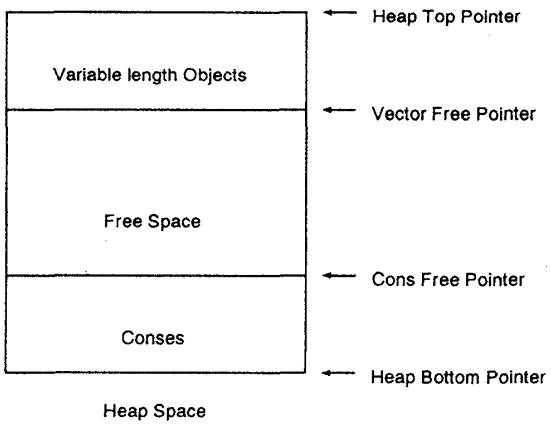

図1: 単一ヒープ2領域記憶管理法

この方法ではCommon Lispで扱うすべてのオブジェクトをヒープと呼ぶ一つの線形空間に置いている。ただし、図1に示すようにヒープ領域を上向きに成長するコンス領域と下向きに成長するベクタ領域とに分割し、オブジェクトの中、Consセル等の固定長オブジェクトをコンス領域に、Vectorやbignum等の可変長オブジェクトをベクタ領域に格納する。これらの領域はCons free pointer、Vector free pointerの2つのポインタで管理される。これら2つの領域が成長し、ポインタが衝突した時点でGCが起動され、各領域がそれぞれコンパクションされる。

3 視覚的モニタリング

Common Lisp 処理系における動的な記憶領域管理をリアルタイムでモニタリングするためには、Lisp プログラムの実行中、オブジェクトがヒープ領域にどの程度格納されているかを常時監視しなければならない。HCl ではヒープ領域を 2 つのポインタによって管理しているため、ヒープ領域の消費量をモニタリングするためにはこの 2 つのポインタを監視すればよい。Common Lisp で扱うオブジェクトに関しては以上の方針をとっている。

また、HCl は MC680X0 をモデルとした仮想マシンを想定し、その上で動作するように設計されている (図 2)。

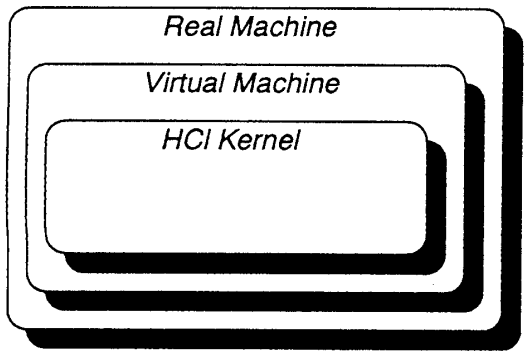

図 2: HCl の動作環境

この仮想マシンでは関数呼出やサブルーチンコールにおけるデータの受け渡しをデータスタックを介して行なうようになっている。すなわち、HCl の内部記憶管理を把握するためには、スタックの使用量も同時にモニタリングする必要がある。仮想マシンでは HCl カーネル等で使われるプログラムスタックと、データの受け渡しに使われるデータスタックの 2 つのスタックを用意しており、動作モニタではそれぞれのスタックのポインタである、Program stack pointe と Data stack pointer の監視を行なっている。

視覚的なモニタリングを行なうために、これまでにあげた 4 つのポインタを監視して X Window を用いて表示させた。視覚的動作モニタを走らせながら Lisp プログラムを実行するため、モニタの動作中はある程度のオーバーヘッドがかかることになる。

なお、実行環境や動作中の処理速度等の結果は当日報告する。

4 おわりに

動的な記憶領域管理のモニタリングは Lisp プログラムの振舞いの理解やデバッグ作業等に役立つものであるが、実行中のプログラムの速度を低下させる要因となる。今後の課題としては、モニタ動作中のオーバーヘッドをいかに小さく抑えるかという問題が考えられる。

参考文献

[1] 山本 強, “小型 Common Lisp 処理系のための記憶領域管理法とその実現”, 情報処理学会論文誌, Vol.31, No.7, July 1990.

[2] D. Kevin Layer, Chris Richardson: Lisp Systems in 1990s. *COMMUNICATIONS OF THE ACM Vol.34*, No.9, Sep. 1991.