

発表概要

実時間処理に適したメモリ管理を行う Lisp 処理系の設計と実装

森 住 大 樹[†] 小 宮 常 康^{††}
 八 杉 昌 宏[†] 湯 浅 太 一[†]

実時間処理を妨げる要因の 1 つとしてごみ集めがあげられる．ごみ集めが一括して行われる場合には，その間プログラムの実行は一時的に停止することとなり，実時間処理の妨げとなる．この問題に対処するため，実時間処理に対応したごみ集めも考慮されているが，Real-Time Specification for Java (RTSJ) では，ごみ集め的方式自体には制限を加えることなく，実時間処理を可能としている．Lisp もごみ集めの機能を持つプログラミング言語の 1 つである．本発表では，RTSJ の方法を取り入れることによって実時間処理にも利用できるように設計，実装した Lisp を紹介する．具体的には，従来のヒープとは別にスコープメモリと呼ばれる実時間処理用のメモリ領域を導入する．スコープメモリはごみ集めにより使用済みのメモリ領域を回収するのではなく，ある程度の大きさでまとめて確保し，必要がなくなったら一度に破棄する．細かいメモリ管理の手間が必要なく，使用中のデータの破棄やメモリリークの恐れもない設計となっており，ごみ集めの利便性を損なうこともない．実装にあたっては，Java により記述された Scheme 処理系である JAKLD と RTSJ を満たす Java 処理系を組み合わせることで，効率良く実装を行うことを可能とした．Lisp に新たに追加する機能は RTSJ の Java 処理系にも含まれるものであり，実装では JAKLD を改良し Java の機能を有効利用した．また，JAKLD を基に作成された L 処理系に対しても，設計と実装を行った．実装された Lisp 処理系は，それ自身十分に動作するものであり，また，より高性能の本格的な処理系を作成する際に参考となりうるものでもある．

Design and Implementation of Lisp System with Memory Management Suitable for Real-Time Processing

HIROKI MORIZUMI,[†] TSUNEYASU KOMIYA,^{††} MASAHIRO YASUGI[†]
 and TAIICHI YUASA[†]

Design and Implementation of Lisp System with Memory Management suitable for Real-Time Processing. Garbage collection (GC) is one of the factors obstruct real-time processing. In case that GC is executed at a time, it stops temporarily execution of program and obstructs real-time processing. For the purpose of dealing with this problem, GC corresponding to real-time processing are devised, but Real-Time Specification for Java (RTSJ) realizes real-time processing without adding restrictions to a method of GC itself. Lisp is one of programming languages with GC. At this presentation, we introduce Lisp which is designed and implemented for real-time processing by taking in the method of RTSJ. Specifically, the memory area called scope memory for real-time processing is introduced apart from the conventional heap. At a scope memory area, GC does not collect used memory, but a certain size memory is secured at a time and deleted at a time if it is not necessary. Because it is designed not to need complicated work for memory management and not to concern about deletion of necessary data and memory leak, the convenience of GC is not spoiled. It made the implementation efficient to combine JAKLD which is a Scheme system described by Java and a Java system of RTSJ. The function newly added to Lisp is included also in a Java system of RTSJ, and the function of Java was available at the implementation by improving JAKLD. We also designed and implemented to L system created from JAKLD. The implemented Lisp system works enough in it self, and it may also be referred when a more highly efficient system is created.

(平成 16 年 3 月 18 日発表)

[†] 京都大学大学院情報学研究科通信情報システム専攻

Department of Communications and Computer Engineering, Graduate School of Informatics, Kyoto University

^{††} 豊橋技術科学大学情報工学系

Department of Information and Computer Sciences, Toyohashi University of Technology