

ANSI C 言語でのガベージコレクション方式の提案

中田 宏† 片岡 信弘†

C 言語を用いて大規模なアプリケーションを記述することは珍しくない。24 時間 365 日ノンストップで動作し続けるアプリケーションも多く、ステップ数で 200k を越える規模のものもある。そのようなアプリケーション内で、動的にメモリの取得、解放を行なう場合、プログラムの設計ミスで不要な領域をいつまでも解放しなかったり、すでに解放した領域を使用中と思い込み誤ってアクセスする例が絶えない。これは、C 言語でのメモリ解放手続きがプログラムに正確に記述されなければならない、記述ミスにより解放を失敗することがあるからである。本論文は、C 言語でのメモリ解放手続きを自動化する方式を提案する。本方式は、ポインタプールと呼ばれる領域に全ポインタのアドレス情報を集め、ポインタからささされている領域を判別することでガベージコレクションを行なう。ポインタをポインタプールに登録するために、ソースコードの変換を行なう。そのため、C プリプロセッサとコードジェネレータの間にトランスレータと呼ぶ変換プログラムを挿入する。また、`malloc()`、`free()` といったライブラリコールを書き換え、本方式のライブラリコールへと変換する。このようにトランスレータとライブラリを追加することで、従来のソースコードを変更することなく C 言語におけるガベージコレクションを実現する方式を提案する。またこの方式を実際に適用し、評価した結果従来の conservative ガベージコレクションよりリソース消費量では不利な面もあるが、より高機能なガベージコレクションが可能であることを検証できた。

Proposal of Garbage Collection Method in ANSI C Language

HIROSHI NAKATA† and NOBUHIRO KATAOKA†

Currently, it is fairly common to develop large scale applications using C language. Some applications run under non-stop situation, and their size can be over 200,000 steps. When allocating and freeing a memory area in such applications, there still exist cases of not freeing unnecessary area caused by mis-design and of accessing already freed area by mistake. This is because the C language itself requires programs to describe the memory free procedure properly and so that it is possible to fail freeing area by mis-programming. This paper proposes an automatic memory freeing method in C language. In this method, we put all the pointer address together in an area called pointer pool, and execute garbage collection by

finding an area which is pointed from a pointer. In order to register the pointer in the pointer pool, translation of source code is executed. To do so, we put a translation program between C preprocessor and code generator. In addition, we replace some memory handling library functions such as `malloc()` and `free()`, by new version of the library. Thus this method can realize the garbage collection in C language by adding the translator and the library functions, without rewriting any source code. We applied this method to evaluate system and confirm that our garbage collection method is same what behind for conservative garbage collection but realise more functionality under the same load.

(平成 10 年 10 月 29 日発表)

† 三菱電機株式会社
Mitsubishi Electric Corporation