

Cプログラムのメモリ使用量表示システム

岡村 拓大[†] 岩澤 京子[‡]

拓殖大学 情報工学科

1. 目的と研究方針

組込みシステムのようにハードウェアに制約がある場合、ソフトウェア開発でメモリの使用量を最小限にする事が大切であるといわれている。

メモリ量を最小限に抑えるためには、ユーザがプログラムのメモリ使用状態を正確に把握し、無駄に使用しているメモリの利用方法を改善できる場所を探せる必要性があると考えた。

そこで、本研究ではソースプログラムに対応させて、使用しているメモリ量を種類ごとに分類してわかりやすく表示させるシステムを開発することにした。

方針として、研究対象とするプログラミング言語はC言語とする。GUI出力はソースプログラムを実行してからオフラインで行う。

2. システム機能

開発したシステムの機能を次に示す。

(1) メモリ情報の測定

測定する情報は、関数呼び出しごとにスタック管理するメモリ量、動的に確保、解放されたメモリ量の3種類である。

(2) 棒グラフによるメモリ使用量の表示

図1のように、メモリ使用量の変化を、時系列を用いて棒グラフで表示する。

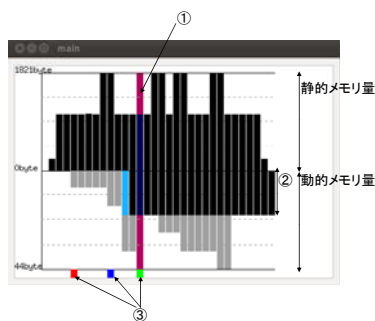


図1 時系列ごとのメモリ使用量

X軸は測定点を時系列で並べその時のメモリ使用量をY軸にbyte単位で示している。上は静的メモリ量、下は動的メモリ量である。

(3) メモリ使用量とソース対応の表示

図1のグラフの任意の棒を指定すると、対話的にソースプログラムを表示する。

free関数を指定すると、対応するallocのグラフ上の位置とソースプログラムを表示する。

図2に例を示す。連結されている動的メモリ量に対しては、すべての確保位置が表示される。

本稿ではこのうち、連結動的メモリの解析について4章で詳しく述べる。

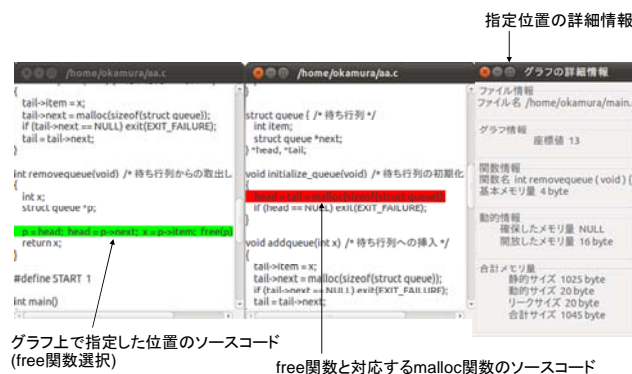


図2 指定位置で表示するソースプログラム

3. システム構成

図3に本システムの全体構成図を示す。

まず、ログ出力生成処理がユーザプログラムのメモリ確保、解放に関する位置を検出し、記録用の関数を呼出すレコードを挿入する。記録用の関数は4種類あり、ログ出力を可能とする。次にプログラムを実行し、ログファイルを生成する。そしてログ解析処理を行い、グラフィック出力する。

Display system of memory usage for C programs
[†]Takuhiko okamura, Graduate school of Information Engineering, Takushoku University
[‡]Kyoko iwawasa, Information Engineering.

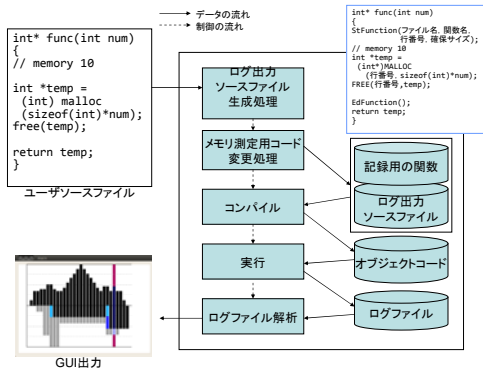


図3 本システムの全体構成

4 連結した動的メモリ検出処理

記録用の関数を用いて、alloc 時に得たアドレス情報をすべて保持し、free の時に指定されたアドレスから連結しているすべてのアドレスを計算して書きこむ。次に ログファイルの生成方法について述べる。

4.1 ログファイルと生成方式

ログ出力ソースファイル生成処理は、次の情報の検出を行う。

- ・ メモリアクセスに関するマクロ定義
- ・ 静的なメモリ宣言
- ・ 動的なメモリの確保と解放

次の4箇所の記録用関数に変更する。

- ・ 関数実行文の先頭
- ・ return の前
- ・ alloc, malloc 系関数・free 関数

図4に示すようメモリ量やアドレス以外に関数名や行番号を引数に加える。

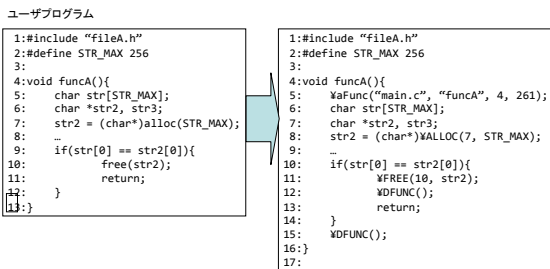


図4 ログ出力ソースファイル

4.2 alloc と free の記録用関数の ¥ALLOC と ¥FREE 処理

連結した動的メモリ量を測定する際、記録用の関数で保持する情報を図5に示す。ソースプログラムから、記録用の関数の ¥ALLOC が呼出されると、ユーザ指定分のメモリ量を alloc で確保する。得られたアドレス値とメモリ量をソース上の位置情報と共に内部データに保持し、同時にログデータに書きこむ。

一方、¥FREE を呼出される時、引数で解放する領域の先頭アドレスから、¥ALLOC が保持した確保した先頭アドレス値からメモリ量分のレコードの検出を行う。アドレス検出では、アドレス値からメモリ量分をアラインメントにより4バイト単位で探す。

ソースプログラム(fileA.c)

```

1: typedef struct strList{
2:   char str[256];
3:   struct strList *next;
4: }strList;
5: Void funcB(){
6:   strList *data;
7:   data = malloc(sizeof(strList));           ①
8:   data->next = malloc(sizeof(strList))      ②
9:   data->next->next = malloc(sizeof(stList)); ③
10:  free(data->next);                          ④
11: }
    
```

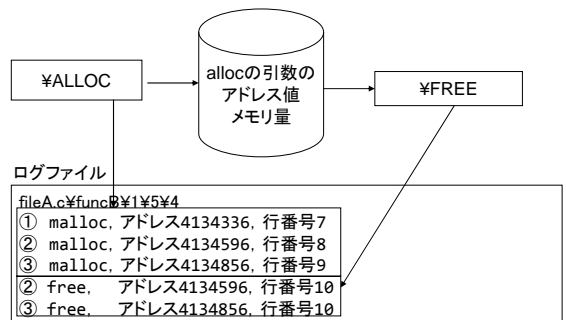


図5 連結した動的情報の取得

4.3 ログファイルの解析

ログファイルからデータを読み込み、タグから内容を判断して適切な構造体にマッピングを行い、各種別の累積されたメモリ量をソースの位置情報と対応付し、GUIで表示する。

5. おわりに

本システムは、C言語ソースファイルを入力とし、ユーザが消費したメモリ使用量を時系列で表示する。システムの開発を行い、メモリ使用量のグラフィック表示と、動的メモリ量の正確な測定が達成できた。実用プログラムで実験していない点も挙げられる。

今後の課題は、実用プログラムでの動作確認である。

参考文献

[1] カーニハン, リッチー: プログラミング言語C 第2版, 共立出版株式会社 (1989)