

アプリケーションによる過剰なメモリ消費を 防ぐための可視化によるユーザ支援

*赤坂俊哉 *櫻井彰人 *岡田有策

1. 動機と目的

アプリケーションが過剰なキャッシングやメモリリークにより、物理メモリ (RAM) 上の占有領域を増やしていけば、スワッピングが頻発し処理速度が低下するだけでなく、当該アプリケーションやシステム全体がクラッシュするような事態にも発展し得る。上記の如くメモリを占有し続けるアプリケーションイベントを、ここではメモリ保持イベントと呼ぶことにするが、このメモリ保持イベントの元となるコードを開発時に完璧に排除することは、少なくとも現状では不可能である。

アプリケーションのユーザにとっては、どのような操作 (アプリケーションイベント) がメモリ保持イベントなのかを特定できることが有益である。というのも、メモリ保持イベントの操作が特定されれば、その操作を使わない、使い過ぎたらアプリケーションを再起動するといった対処が出来るからである。しかし、OS 等のシステムソフトウェアでは、アプリケーション層の詳細な情報が見えないため、これらのソフトウェアにメモリ保持イベントを特定させることは難しい。一方で、個々のアプリケーションにメモリ保持イベントを特定するような機能を求めるのも非現実的である。

そこで本研究では、アプリケーションのユーザ自身がメモリ保持イベントを特定できるような方策を考えることにした。具体的には、次の仮説を検証することを目的とした。すなわち、「各プログラムの RAM 占有量が常時可視化されていれば、ユーザは各イベントとメモリ消費との関係を把握できるようになり、ひいてはメモリ保持イベントを特定できるようになる」という仮説である。

2. 関連研究

メモリリークの検出を支援する研究として、例えば[2], [3]などが挙げられるが、これらはともに特定の言語で書かれたプログラムのデバッキングやチューニングを支援することを目指した研究である。[1]など、メモリを可視化する試みも以前から行われているが、それらの研究も、やはりプログラムのデバッキング支援を目的としている。メモリリークの検知支援にしても、メモリの可視化にしても、アプリケーションのユーザを対象とした研究は少ない。本研究は、メモリ保持イベントの (デバッキングではなく) 誘因の特定を目的とし、それをユーザに行わせようとする試みである。

3. 本研究の位置付け

本研究の目的は、「メモリの可視化はメモリ保持イベントの特定を支援できる」という可能性を確かめて、今後の研究の基礎を築くことにある。それゆえ本研究では、被験者実験における被験者を「メモリ管理に対する背景知識を持つ者」とし、有効性の確認に有利に働くような被験者集団を敢えて選んだ。メモリ提示方法についても、時間軸とメモリ消費量軸を持つグラフ表示としたが、他の提示方法との比較が行われているわけではない。「知識の無いユーザでも可視化されたメモリを利用できるのか」、「どのような提示方法が最適なのか」などは、有効性が示された後の研究課題と位置づけ、本研究では議論の対象外とした。

4. 実験

「1. 動機と目的」で述べた仮説の検証を目的とした被験者実験を行った。実験にあたり、メモ

*慶應義塾大学理工学部管理工学科

り保持イベントを起こすアプリケーションとして、以下の機能を持つ簡単な文書編集ソフト（図1上方のウィンドウが編集画面）を開発した。

- 文書の保存
- 文書のプリントアウト
- テキストの切り取り/コピー/貼付け
- オブジェクトの切り取り/コピー/貼付け

これらの中で、オブジェクトの貼付け操作がメモリ保持イベントとなっており、オブジェクト貼付け時に、バッファとしてRAMを獲得して、それをそのまま占有し続けるような作りになっている。

被験者には、このソフトを使って文書編集作業を行わせた後で、メモリ保持イベントを選ぶ選択式問題を回答させた。被験者は、可視化されたメモリを見ながら作業するグループ（可視化グループ）と、何も見ないで作業するグループ（非可視化グループ）に分かれており、それぞれのグループにおいて上記問題に正答できた人数を集計した。

メモリ可視化用のウィンドウ（図1下方のウィンドウ）は、常に画面下端に位置し、出来る限り他のウィンドウを妨げないようにしている。ウィンドウの縦軸をRAM消費量、横軸を時間軸（右端が最近時）とし、各プロセスのRAM消費量の時間変化が折れ線グラフ（図1には3本）で表示される。あるプロセスが終了すれば対応するグラフも画面上から消える。縦軸の幅は80MB相当に固定されているが、最もRAMを消費しているプロセスのグラフが常に表示されるよう、縦軸の上下端の値が動的に調整される。

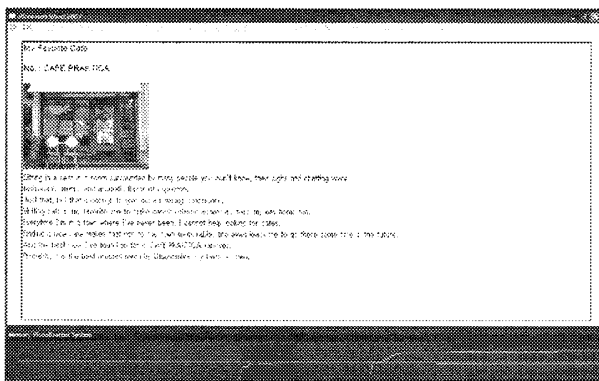


図1. 文書編集ソフトとメモリ可視化ソフト

被験者には、スワッピング、キャッシング、メモリリーク等の知識を問うテストを事前に受験させ、全問正解者のみを採用した。採用された被験者は20代の大学生・大学院生から成る13名であり、うち7名を可視化グループ、残り6名を非可視化グループとした。

5. 結果

メモリ保持イベント（オブジェクトの貼付け）を選択肢の中から選ぶ問題に対する可視化・非可視化グループの正答者数は、次の通りであった。

可視化グループ：7人中3人

非可視化グループ：6人中0人

正答者の割合からしても可視化グループの方が良い成績だと言えるが、誤答の質にもグループ間で差が見られた。選択肢のうち、非可視化グループは全員「分からない」を選択しているのに対し、可視化グループの誤答者4人のうち3人は「オブジェクトのコピー」を選択しており、「オブジェクトのクリップボード操作のいずれか」というところにまで絞り込めていたことが分かる。

6. 結論

メモリリークや過剰なキャッシング（メモリ保持イベント）を引き起こすような操作をユーザの側から特定する際に、メモリの可視化がユーザを支援し得るという可能性を確認することができた。

参考文献

- [1] Ralph E. Griswold and Gregg M. Townsend, The visualization of dynamic memory management in the Icon programming language, Technical Report 89-30, Department of Computer Science, University of Arizona, December 1989.
- [2] R. Hastings and B. Joyce, Purify: fast detection of memory leaks and access errors, Proceedings of the Winter 1992 USENIX Conference, Berkeley, CA, 1991, pp. 125-36.
- [3] W. De Pauw and G. Sevitsky, Visualizing reference patterns for solving memory leaks in Java, Proceedings of the 14th European Conference on Object-Oriented Programming, 1999, pp. 116 - 134.