JavaScript 処理系 eJS の Mbed マイコンへの移植と静的なデータのフラッシュメモリへの配置

1210339　近森凪沙　【 ソフトウェア検証・解析学研究室 】

1　はじめに

組み込みシステム向けの JavaScript 処理系 eJS[1] は，これまで UNIX 上で開発しており，動作実験も UNIX 上で行っていた．そこで本研究では，eJS を実際の組み込みシステムで使用される Mbed[2] 対応のマイコンで動作させる．本研究では 256 KB の RAM を持つ FRDM-K64F で eJS が動作するようにビルドする仕組みを作る．そのとき，JavaScript プログラムが使用するメモリ領域である JavaScript ヒープを大きくできるように工夫し，更に JavaScript ヒープ内で，VM 起動時に生成されていた静的データのうち，特に文字列オブジェクトをフラッシュメモリ（以下，フラッシュ）に配置し，JavaScript ヒープの利用可能な領域を増やす．

2　eJS の移植

既存の eJS では，中間コードである OBC にコンパイルし，コンパイル結果が記述された OBC ファイルをファイルシステムから読み込む．しかし，組み込みシステムでファイルシステムが利用できるとは限らない．そこで，実行するプログラムは一つだけという前提のもと，OBC を配列に格納して VM に直接埋め込むよう eJS を改変する．配列はフラッシュに配置する．

既存の eJS では，malloc を使用して JavaScript ヒープのメモリ領域を確保する．空き領域全体を JavaScript ヒープとしたいが，malloc が取得できる最大量は.bss セクション等の他のデータ領域の大きさによって変わる．また，malloc の管理領域によるオーバヘッドが発生する．そこで，リンカスクリプトを改変して JavaScript ヒープ専用の領域（.jsheap セクション）を生成し，.jsheap セクション全体を JavaScript ヒープとする．eJS で.jsheap セクションから JavaScript ヒープを確保するようにすることで，利用できる領域の最大量を自動的に取得することができ，malloc の管理領域も不要となる．

3　静的なデータのフラッシュメモリへの配置

JavaScript ヒープが小さいと，メモリ不足によりプログラムの実行が停止してしまうことがある．また，GC の実行回数の増加により VM の実行が遅くなる．そこで，文字列オブジェクトをフラッシュへ配置することで，文字列オブジェクトが占めていた領域を他のオブジェクトが使用できるようにする．本研究では，VM 起動時に内部的に作られる文字列オブジェクトと，プログラム中に存在する文字列リテラルやオブジェクトのプロパティ名に対応する文字列オブジェクトをフラッシュに配置する．

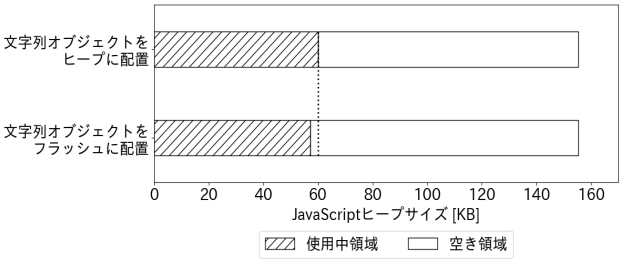

図 1　JavaScript ヒープの内訳

4　評価

OS は Mbed OS 5，ライブラリは組み込みシステム向け標準 C ライブラリである Newlib 3.1.0 を使用する．

まず，JavaScript ヒープを malloc を介して取得した場合と，.jsheap セクションから取得した場合のサイズを調査した．malloc で取得するバイト数は，試行錯誤により取得可能な最大値を探した．malloc を介して取得した場合は最大 151,532 B に対し，.jsheap セクションから取得した場合は 155,264 B の JavaScript ヒープを作ることができ，約 4 KB 大きくなった．

次に，文字列オブジェクトをヒープに配置した場合とフラッシュに配置した場合の，VM 初期化直後における JavaScirpt ヒープの空き領域を比較した．ベンチマークは，SunSpider の一部を eJS 用に改変したものを利用した．ベンチマークの一つである，access-binary-trees の JavaScript ヒープの内訳を図 1 に示す．文字列オブジェクトをフラッシュに配置したことで，空き領域が約 3 KB 増加した．他のベンチマークも同じ傾向であった．これにより GC の実行回数が減少し，実行が最大約 30％高速になったプログラムがあった．フラッシュへのアクセスによるオーバヘッドは最大約 2％だった．

5　まとめ

本研究では，eJS を Mbed 上で動作するようにビルドする仕組みを構築した．これにより，Mbed 上で eJS を動作させることが可能となった．また，malloc を介さず.jsheap セクションを使用して JavaScript ヒープを大きくし，更に文字列オブジェクトをフラッシュに配置することで，空き領域が増加した．

参考文献

[1] T. Kataoka, T. Ugawa, and H. Iwasaki. A Framework for Constructing JavaScript Virtual Machines with Customized Datatype Representations. SAC'18, pp.1238–1247, 2018.

[2] Mbed. https://os.mbed.com/, 2020.