

SWEST20 s3c 「mruby
の省メモリ化について考える」
その後のご報告

2019/09/05 SWEST 21
山根ゆりえ

agenda

- SWEST20 セッションの目的
- SWEST20セッションの概要
- 現時点での達成状況
- 今後について

SWEST20セッションの目的

- mrubyでマイコンプログラミングできるようになることにより、C言語を知らないユーザー層にもアプローチできるのではないかと仮定
- mrubyはC言語に比べ大量のRAMを消費するため、RAM容量の多いマイコンを使う必要がある
 - RAM容量の多いマイコンはArduinoより高価なため、初心者層に受けなかった。
 - mruby初期リリース当時はもう少しRAMの価格が安くなり、RAM容量の多いマイコンも安価に入手できると想定していた
 - 思ったほど下がらなかったため、RAM使用量を減らすことによりArduinoと同価格帯のマイコンでも動くようにする必要があると判断した

SWEST20セッションの概要

- mrubyは初期化時にRAMを多く消費するが、本当にRAMである必要があるのかを検討した結果、ROMに置きそうなものが結構あることが判明した
- mruby作者のまつもとゆきひろさんが参加する&組込み技術者が集まる＝絶好の機会ではないか！
- 構想や課題などを紹介し、まつもとさんやセッション参加者とディスカッションした
- 基本的な方向性は合っていると確認できた

SWEST20での課題

- SWEST20開催時での課題は以下のとおり
 - RClassのROM化
 - irepのROM化
 - mruby 2.0 (バイトコードが変更) 対応
 - 作業中にリリースされたため追加対応が必要となった
 - khash.hをtree等書き換える
 - irepを削る

セッション後の進捗

- RAMとROMの使い分けの検討
 - ROM
 - 関数テーブル（初期化時のもの）
 - mruby VMスタック
 - irep
 - range_edges
 - mrubyオブジェクト（RBasic構造体）
 - 初期mrb_state構造体
 - 実行時にRAMにコピーして利用
 - RAM
 - htable構造体
 - 関数テーブル（実行時に追加されるもの）
 - iv_tbl構造体
 - mrb_context構造体

セッション後の進捗

- TLSFによるメモリ管理
 - mrubyのGCはmrb_mallocを使用して確保した領域のみ対象
 - ROM化対応によりmrubyのメモリ管理から外れた部分（GC対象とならない）を区別するために導入
- mruby 2.0対応
 - Classクラス(RClass)のivのsegment list化
 - Hashクラス(RHash)のhashテーブル化
- RBasic構造体のflagsの可変対応

現時点での達成状況

- RClassのROM化
 - RClassに限らず、標準的なクラスの実行時に変更不要なところはROMに置いた
- irepのROM化
- mruby 2.0 (バイトコードが変更) 対応
- ~~(khash.hをtree等~~に書き換える
 - mruby本体のバージョンアップにより不要となった
- irepを削る
 - 全てROMに置くことにより対応不要となった

今後について

- mruby本体への実装に向けて作業中です
 - nvgen(ROM化するプログラム)をmrbgemsにする
 - mruby本体へのフィードバック