

mruby/cによる マルチプログラミングの提案

田中 和明
九州工業大学

概要

- マルチプログラミング
- mrubyとmruby/c

自己紹介

自己紹介

- 田中 和明（九州工業大学 情報工学部）
- 教育
 - プログラミング、データ構造とアルゴリズムなどの講義を担当
- 研究
 - 組込みシステム（mruby）
 - ブロックチェーン技術（をIoTで利用する）
 - LPWA、光学測量に関する共同研究

mrubyの開発

- 2010年度～2012年度
 - 経済産業省 地域イノベーション創出研究開発事業でmrubyを開発（ネットワーク応用通信研究所、福岡CSK、九州工業大学）
- 2015年度～
 - 島根県 しまねソフト研究開発センターとの共同研究でmruby/cを開発
- 2016年度
 - NEDO 次世代スマートデバイス開発プロジェクトでmruby用IDEを開発（九州工業大学、SCSK九州、有明高専）
- 2017年度～2018年度
 - 経済産業省 戦略的基盤技術高度化支援事業（サポイン事業）でmrubyを使った省電力無線モジュールを開発（Braveridge、九州工業大学、SCSK九州、ILC）

マルチプログラミング

マルチプログラミング

- 複数の機能を実装する
↓
- プログラムが複雑になる
 - 異なるタイミングでそれぞれの機能を動かす
 - 機能ごとに実装を分離したい
 - セキュリティ
 - 保守性
- マルチプログラミング

例

- 赤色LEDを1秒間隔で点滅させる

```
while true do
  output 0,1
  sleep_ms 1000
  output 0,0
  sleep_ms 1000
end
```

output : GPIO出力

sleep_ms : スリープ

例



- 赤色LEDを1秒間隔で点滅させる
- 緑色LEDを0.1秒間隔で点滅させる

- シングルプログラミングの場合は？
- マルチプログラミングの場合は？

シングルプログラミングの例

もう少しスマートな方法があるかもしれませんが、、、

```
t = 0
while true do
  # 緑色LED
  if t%2 == 0 then
    output 7, 0
  else
    output 7, 1
  end
end
```



```
# 赤色LED&リセット
if t == 10 then
  output 0, 1
elsif t >= 20 then
  t = 0
  output 0, 0
end
sleep_ms 100
t += 1
end
```

マルチプログラミングの例

- 割り込みを使う？
- スレッドを使う？
- マルチタスク（プロセス）を使う？

- もっと簡単に実装できないか？

mruby/c の マルチプログラミング

```
while true do
  output 0, 1
  sleep_ms 1000
  output 0, 0
  sleep_ms 1000
end
```

```
while true do
  output 7, 1
  sleep_ms 100
  output 7, 0
  sleep_ms 100
end
```

mrubyとmruby/c

mruby

- mruby
 - Rubyを軽量化したもの
～Rubyの開発効率を組み込みソフト開発へ
 - Rubyとの互換性を重視
～Ruby技術者が組み込み開発できる
 - 消費メモリは 400KB 程度

mrubyの利用事例



ネットワークルータ(IIJ社)



ゲーム(スクウェアエニックス、SidMeier's)



Apacheモジュール(OSS)



ETロボコン(アフレル社、ほか)



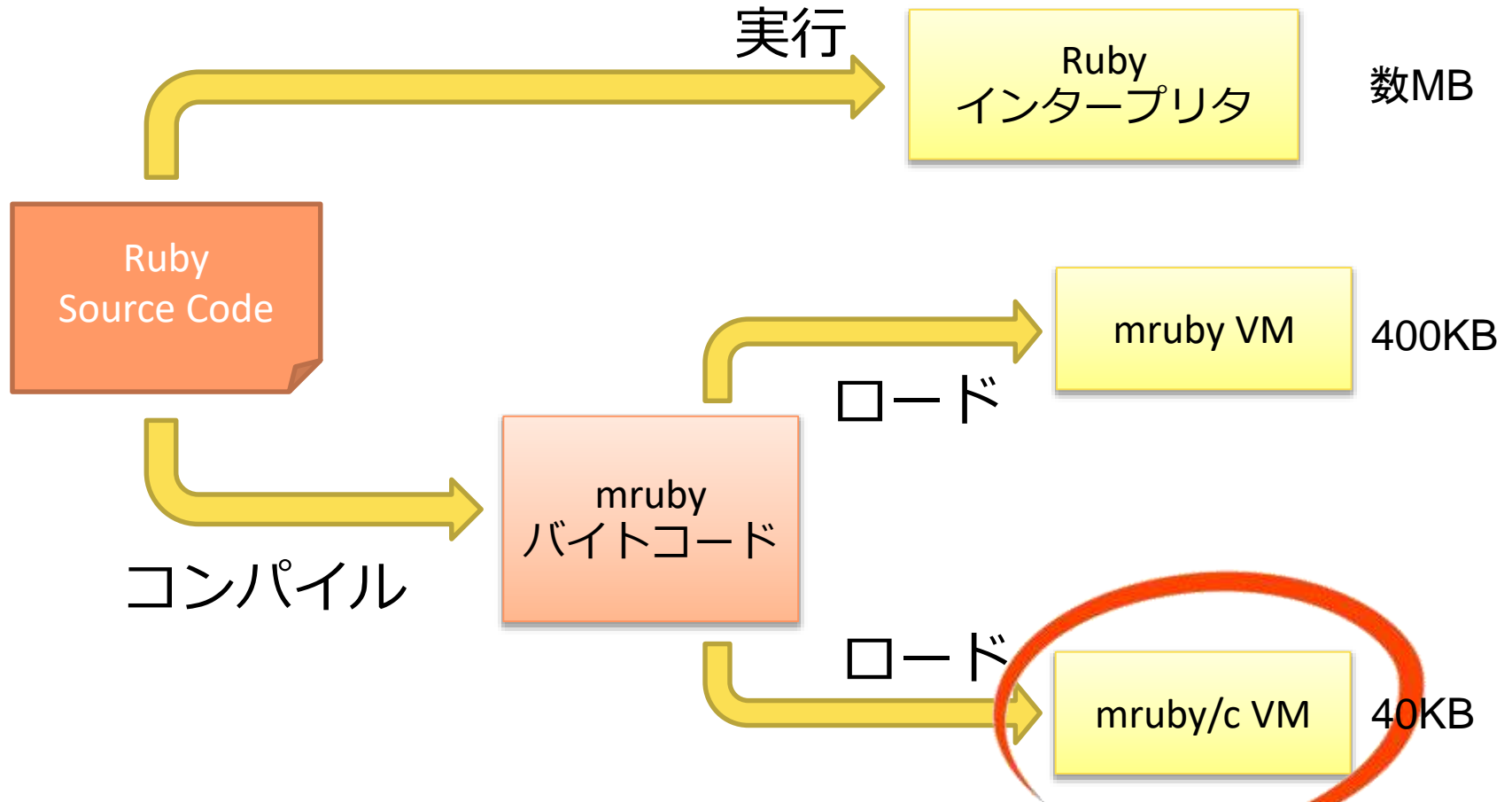
GUIライブラリ(ILC社)

mruby/c

- mruby/c
 - mrubyをさらに軽量化した～40KB程度
ワンチップマイコンでの動作を想定する
 - Rubyとの互換性よりも、小型化を重視
最小限のクラスライブラリ
 - OSを必要としない

Ruby, mruby, mruby/c

実行時に必要なメモリ



mruby/c のポイント

- OSを必要としない
↓
- OSが提供する機能を持つ
 - マルチプログラミング
複数プログラムのコンカレント実行
 - メモリ管理
 - 排他制御
- ユーザプログラムとして実行する

mrubyのバイトコード

- コンパイル後に生成される

```
while true do
  output 0,1
  sleep_ms 1000
  output 0,0
  sleep_ms 1000
end
```



```
1 000 OP_JMP          015
2 001 OP_LOADSELF    R1
2 002 OP_LOADI       R2      0
2 003 OP_LOADI       R3      1
2 004 OP_SEND        R1      :output 2
3 005 OP_LOADSELF    R1
3 006 OP_LOADI       R2      1000
3 007 OP_SEND        R1      :sleep_ms 1
4 008 OP_LOADSELF    R1
4 009 OP_LOADI       R2      0
4 010 OP_LOADI       R3      0
4 011 OP_SEND        R1      :output 2
5 012 OP_LOADSELF    R1
5 013 OP_LOADI       R2      1000
5 014 OP_SEND        R1      :sleep_ms 1
1 015 OP_LOADT       R1
1 016 OP_JMPIF       R1      001
1 017 OP_LOADNIL     R1
1 018 OP_STOP
```

mruby VM

- VMがバイトコードを実行する

```
loop do
  instruction = Fetch(program)
  Execute(instruction)
end
```

mruby/c VMのアイデア

- 複数のプログラム（複数のバイトコード）を実行する

```
loop do
  for i=1 to N do
    current_program = program[i]
    instruction = Fetch(current_program)
    Execute(instruction)
  end
end
```

実行効率の問題

- 実行効率が悪いので、
いくつかのバイトコードを実行する

```
loop do
  for i=1 to N do
    current_program = program[i]
    for j=1 to M do
      instruction = Fetch(current_program)
      Execute(instruction)
    end
  end
end
end
```

実行の公平性

- プログラム間で実行時間が公平ではない

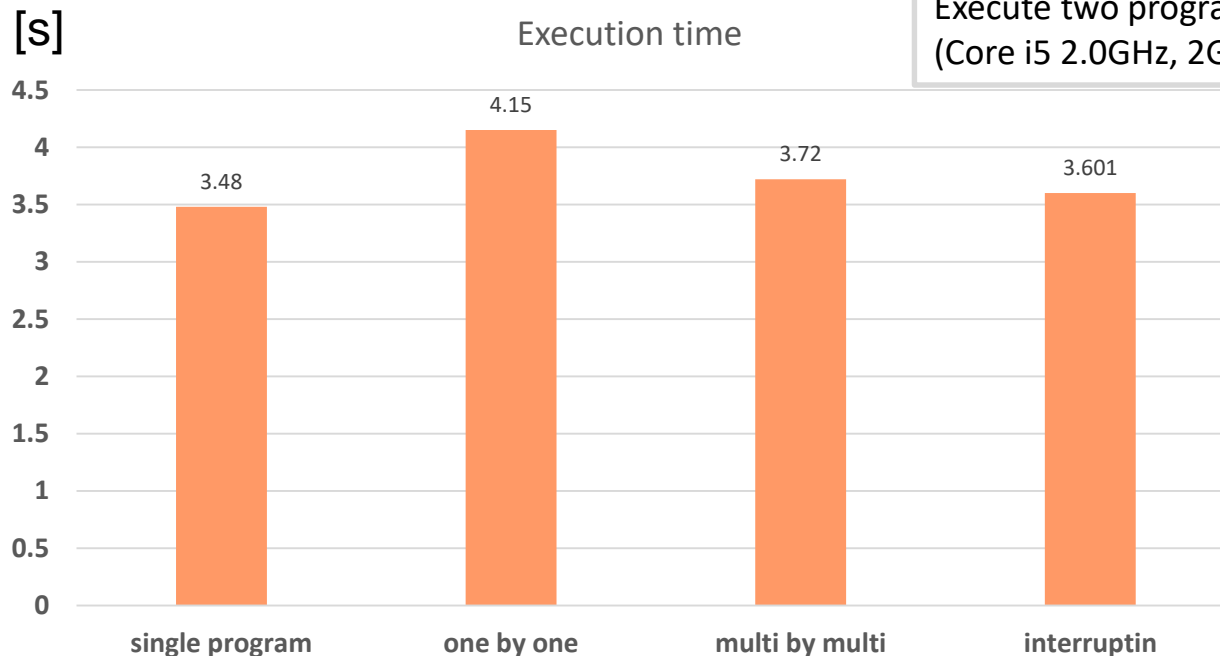
```
loop do
  for i=1 to N do
    current_program = program[i]
    start_time = GetCurrentTime()
    while GetCurrentTime() < (start_time + TIME_SLICE) do
      instruction = Fetch(current_program)
      Execute(instruction)
    end
  end
end
end
```

mruby/c の実装

- タイマー割り込みが使える場合は、
割り込みでプログラム切り替える

Result

- Execution time
 - Calculate 32th Fibonacci number using recursion



Single program execution time is 1.74[s].
Execute two programs by concurrent execution.
(Core i5 2.0GHz, 2GB memory, Ubuntu 14.04)

資料

Resources

- mruby

`http://mruby.org/`

`https://github.com/mruby/mruby`

- mruby/c

`http://www.s-itoc.jp/`

`https://github.com/mruby/mruby`