

Fukuoka.rb #397

2025.03.19 於 SmartHR 九州支社



Fukuoka.rb

```
.where(location: "SmartHR")  
.first
```

RubyKaigiで通らなかった方の プロポーザル

2025.03.19 Fukuoka.rb なんと397回

近藤 うちお (@udzura)

SmartHR プロダクトエンジニア

自己紹介

近藤 宇智郎 (@udzura)

株式会社 SmartHR プロダクトエンジニア

労務基本機能担当

RubyKaigi 2025 Speaker!!

好きなRubyメソッド: `String#unpack`

好きなSmartHRの機能: 申請



RubyKaigi

プロポーザルの様子です



Towards the Possibility of Implementing "componentize-rb"

Speaker: Uchio KONDO **Format:** Regular Session **Track:** General

Updated: about 2 months ago

NOT ACCEPTED

 0 comments



Running ruby.wasm on Pure Ruby WASM Runtime

Speaker: Uchio KONDO **Format:** Regular Session **Track:** General

Updated: about 2 months ago

ACCEPTED

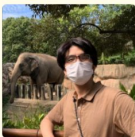
 0 comments

もう一つのプロポーザル ...



前提知識



[< Back](#)



Uchio KONDO

  @udzura

A Rubyist from Fukuoka. An Infrastructure Engineer at Mirrativ, Inc.(a live-streaming platform in Japan). Loves Ruby, mruby, Rust, eBPF, WASM, LLVM, MP4 Media File Format and other binaries. A speaker of Several RubyKaigis, a RubyKaigi 2019 Local Organizer, and a co-translator of o'reilly's "Learning eBPF" (Japanese edition).

EN

An mruby for WebAssembly

The author is developing "mruby/edge", a specialized Ruby runtime for WebAssembly (WASM).

One of the reasons the author started to build a new runtime is that he aimed to create a smaller wasm binary than CRuby's, which is 18 MB (observed from 3.3.0 build). A reduction in binary size addresses the diverse use cases of WASM.

Additionally, mruby/edge associates Ruby methods with corresponding WASM-exported functions. This is in contrast to the implementations of Ruby.wasm and mruby-wasmized-by-emscripen, which only support top-level script execution.

To accomplish these objectives, the author decided to build a VM from scratch, utilizing mruby's elegant bytecode spec for its instruction set. This places mruby/edge within the "mruby family" as a new member.

The talk will cover its internal technologies, challenges, mruby VM and WASM insights, and future prospects, including support for WASM Component Model in mruby/edge.

Presentation Material

 Slides >

mruby/edge とは何だったのか

- ・Rustで書いたmruby VM



- ・Rustで全部書いてるので、WASMにコンパイルするのが簡単

プロポーザルの要約：

mruby/edgeをComponent Model 対応させたり色々の話

プロポーザルの要約：

mruby/edgeをComponent Model
対応させたり色々の話

WASM Component Model ?

- ・WASM同士や、WASMと外部プログラムをいい感じに連携するための仕組み(極めて大雑把な説明)
- ・WASM Componentを組み合わせて別のComponentを作ったりもできる

Gemini に説明を書かせたよ

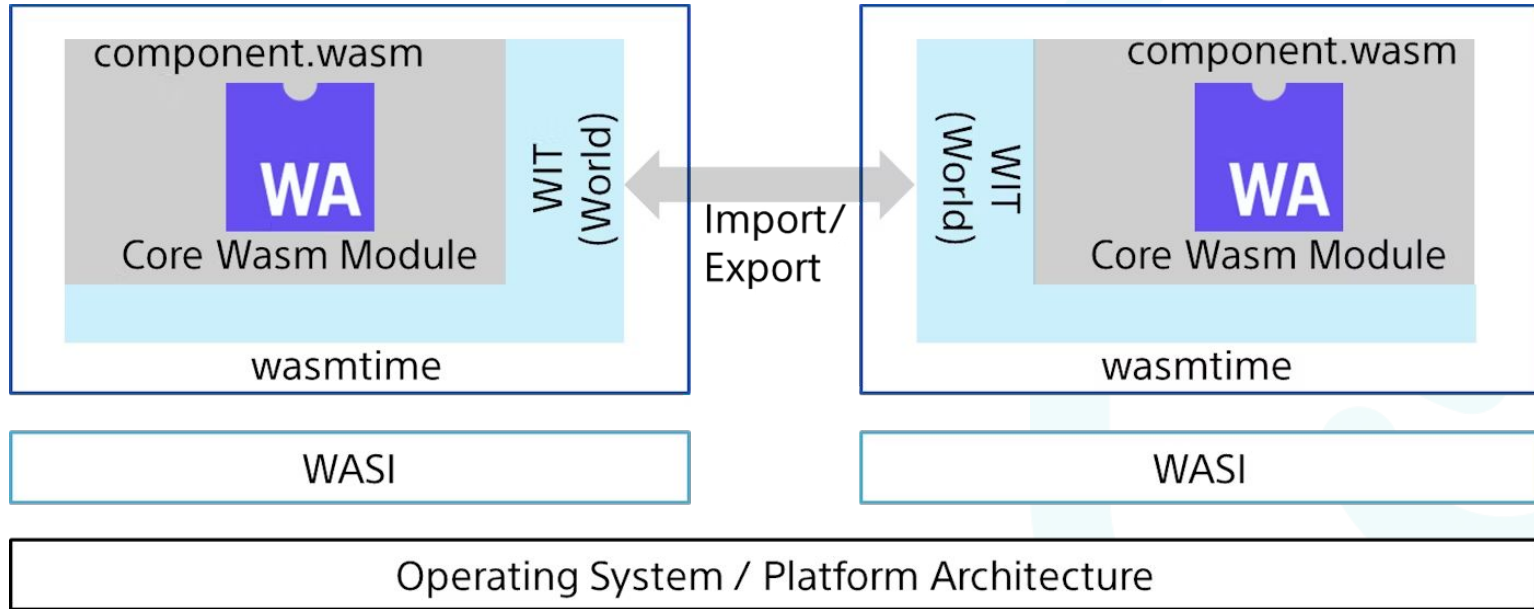
WebAssembly Component Modelとは

WebAssembly Component Modelは、Wasmモジュールを相互運用可能な形で組み合わせるためのアーキテクチャです。従来の Wasm Core仕様では、モジュール間の相互運用性が乏しく、実装者が注意深く開発する必要がありました。Component Modelは、この課題を解決するために、高水準な標準インターフェースを定義することで、Wasmモジュールの再利用性と相互運用性を向上させます。...

例: WASI と Component Model

- ・WASI = WasmでOSの機能を使う規約的なやつ
- ・WASI preview 1 (0.1) は素朴なimport関数
- ・WASI preview 2 (0.2) は...

Component のインタフェースとして定義されている(WIT)
どれをサポートするか、実装は正しいか、など対応しやすく



WASI と Component Model

- WITという形式で

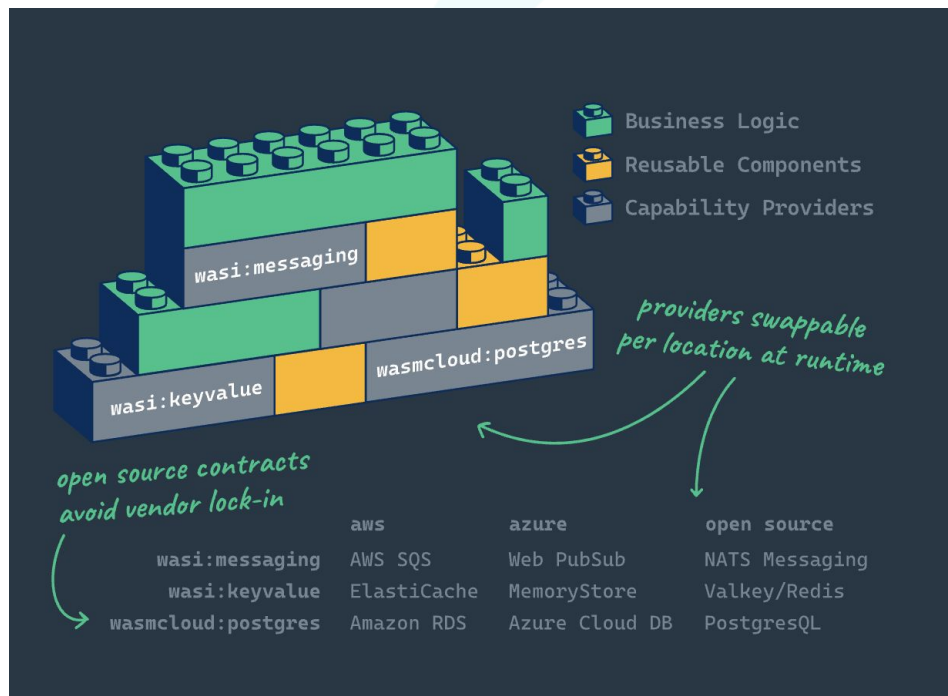
インタフェースを定義している

- protobuf/gRPCのような側面

```
1 package wasi:random@0.2.4;
2 /// WASI Random is a random data API.
3 ///
4 /// It is intended to be portable at least between Unix-family platforms and
5 /// Windows.
6 @since(version = 0.2.0)
7 interface random {
8     /// Return `len` cryptographically-secure random or pseudo-random bytes.
9     ///
10    /// This function must produce data at least as cryptographically secure and
11    /// fast as an adequately seeded cryptographically-secure pseudo-random
12    /// number generator (CSPRNG). It must not block, from the perspective of
13    /// the calling program, under any circumstances, including on the first
14    /// request and on requests for numbers of bytes. The returned data must
15    /// always be unpredictable.
16    ///
17    /// This function must always return fresh data. Deterministic environments
18    /// must omit this function, rather than implementing it with deterministic
19    /// data.
20    @since(version = 0.2.0)
21    get-random-bytes: func(len: u64) -> list<u8>;
22
23    /// Return a cryptographically-secure random or pseudo-random `u64` value.
24    ///
25    /// This function returns the same type of data as `get-random-bytes`,
26    /// represented as a `u64`.
27    @since(version = 0.2.0)
28    get-random-u64: func() -> u64;
29 }
```

イメージ画像

wasmCloudの説明より



Component Model is 難しい...

- ・要するにシンプル組み合わせでアプリを作りやすくなるよ！
ということではあります... 過渡期なので情報もなく...
- ・Component Modelが普及したら言語問わずWASMを作れて
組み合わせることが簡単・安全になるよ！という理解でOK

まあ難しい話は
置いときましょう


mruby/edgeの 進捗報告

RubyKaigi 2024 段階

これは動く

```
1  def fib(n)
2      if n < 1
3          return 0
4      elsif n < 3
5          return 1
6      else
7          return fib(n-1)+fib(n-2)
8      end
9  end
```

2024 時点でのマイルストーンというか ...

- ・ぶっちゃけ「Demo用のfib() が動く」がゴール...
- ・でも、WASMらしく動かすための機能は作っておいた 
- ・関数のimport/export対応
- ・最低限文字列の取り扱い

時は流れ



最近実装したもの

まず、VM

書き換えをした

- ・mruby/c 1.2の実装を
ガン見してRust化した

2025-01-03

■ mruby/edge のVMを書き換えた

明けましておめでとうございます！

タイトルの通り。mruby/edge のVMを、mruby/c 1.2 相当の内容に書き換えた。

mruby/c 1.2 の実装は以前軽く読んだ。



ローファイ日記

id:udzura



Hatena Blog

mruby/c 1.2 のコードを読んで理解するmruby VMの実装

mrubyファミリ (組み込み向け軽量Ruby) Advent Calendar 2024、23日目の記事です。
メリークリスマス！ qiita.com mrubyファミリAdvent Calendar、今年は1日目からyhar
aさんによる衝撃的な記事がありました。yhar.jp なぜ衝撃的だったかという、僕...

2024-12-23 07:21 ★★★★★ 1 user

udzura.hatenablog.jp

PRはめっちゃでかい。 +2,605 -1,376 だそうです。

Rewrite the whole VM by udzura · Pull Request #1 · udzura/mrubyedge

PR
rite the whole VM

220 reviews 44 files +205 -129 6 100%

avg 5.0/5 ⬆ 33 commits

github.com

<https://udzura.hatenablog.jp/entry/2025/01/03/192950>

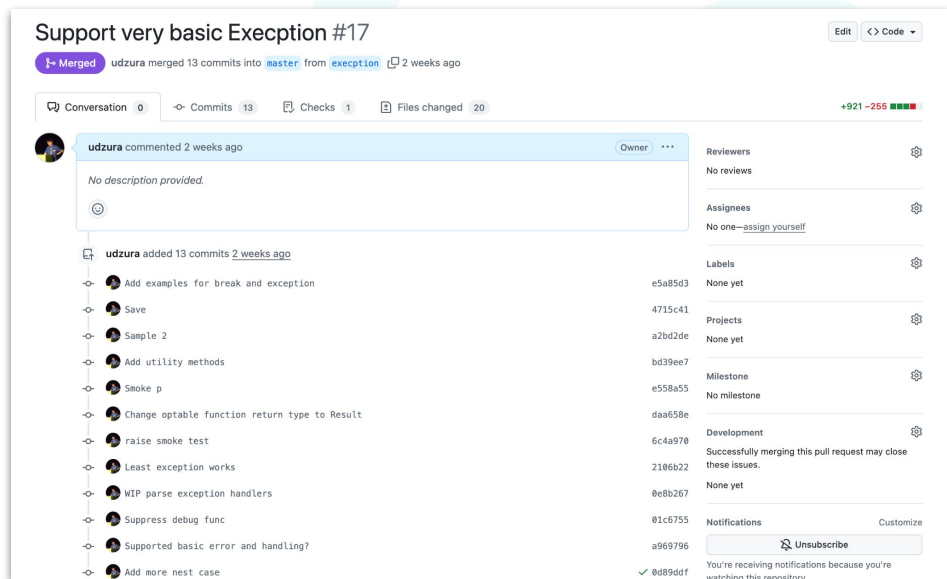
mruby/edge の最近実装した機能

- クラスの定義ができるようにした
- クラスでのインスタンス変数の利用ができるようになった
- クラス定義で `attr_(reader|writer|accessor)` が使えるようになった
 - Cではやりづらい、ネイティブコードでの動的メソッド定義がRustでは普通にできるので天国のようです
- グローバル変数の利用ができるようになった
- ブロック引数の利用をサポートし、一部メソッドを実装した (`Integer#times`, `Array#each`)
- ある程度正しく動くクロージャールを実装した。mruby/c では動かないケース (Qiitaの外部記事に飛びます) も動くようにしているぞ
- WASM instance内部でVMインスタンスを保持するようになった。つまり状態を持てるようになった。今までは関数呼び出しごとにVMを初期化していた
- オフセットを固定したメモリ領域をWASM外に公開できるようにした。 `SharedMemory` というクラスを導入している
 - オフセットを固定することで、WASM外でデータをメモリ内部に書き込み、WASM内部で読み取るというユースケースに対応した
 - 具体的にはこういう用途なんだけど、わかりづらいので頑張ってドキュメントを書く...
- メモリ上のオブジェクトやり取りがしやすいように、 `Array#pack`, `String#unpack` のうち最低限の実装をサポートした
- 配列をサポートした
- Hashをサポートした
- `==` による比較が一通り動くようになった

この後実装した機能

- 例外の実装
- まあまあ
ちゃんと動く？
- setjmp使ってません

(Resultベース)



つまり？

- ・ほぼRubyになった 
- ・なんなら、mruby/c よりちょっとRubyめいている

mruby/c では

このコードは動かない

(よね？はず)

```
1  def do_time_block
2    result = 0
3    ->(i) {
4      result += 100
5      3.times do |j|
6        result += 200
7        puts "result = #{result}"
8      end
9      puts "result = #{result}"
10   }
11 end
12
13 def do_times
14   3.times(&do_time_block)
15 end
```

細かい話

- ・mruby/c はProcオブジェクトを作った環境が捨てられてしまったら、もう参照できない。

- ・環境をキャプチャする分のメモリを節約するため、捨てている

※ エスパー含む感想です

(レジスタをスタックが積まれるごとにシフトしてるイメージ)

- ・これは、mruby/edgeでは動くようにした。

<https://qiita.com/HirohitoHigashi/items/14ffd29e1c23e6989191#%E3%82%AF%E3%83%AD%E3%83%BC%E3%82%B8%E3%83%A3> 参照

今後やりたいこと

- ・普通のRubyの機能実装(せめてmruby/c 3.0ぐらいには)
 - ・Module, 各種メタプロ, キーワード引数, break あたり優先
 - ・何より標準ライブラリをね...
 - ・逆にTimeとかRandomとかサポートしてもいいと思ってる
- (WASIビルドfeatureには対応してる)

コントリビュータ募集中です

Demoがてら： wasmbotsの話

wasmbots とは...

Gemini、頼むよ

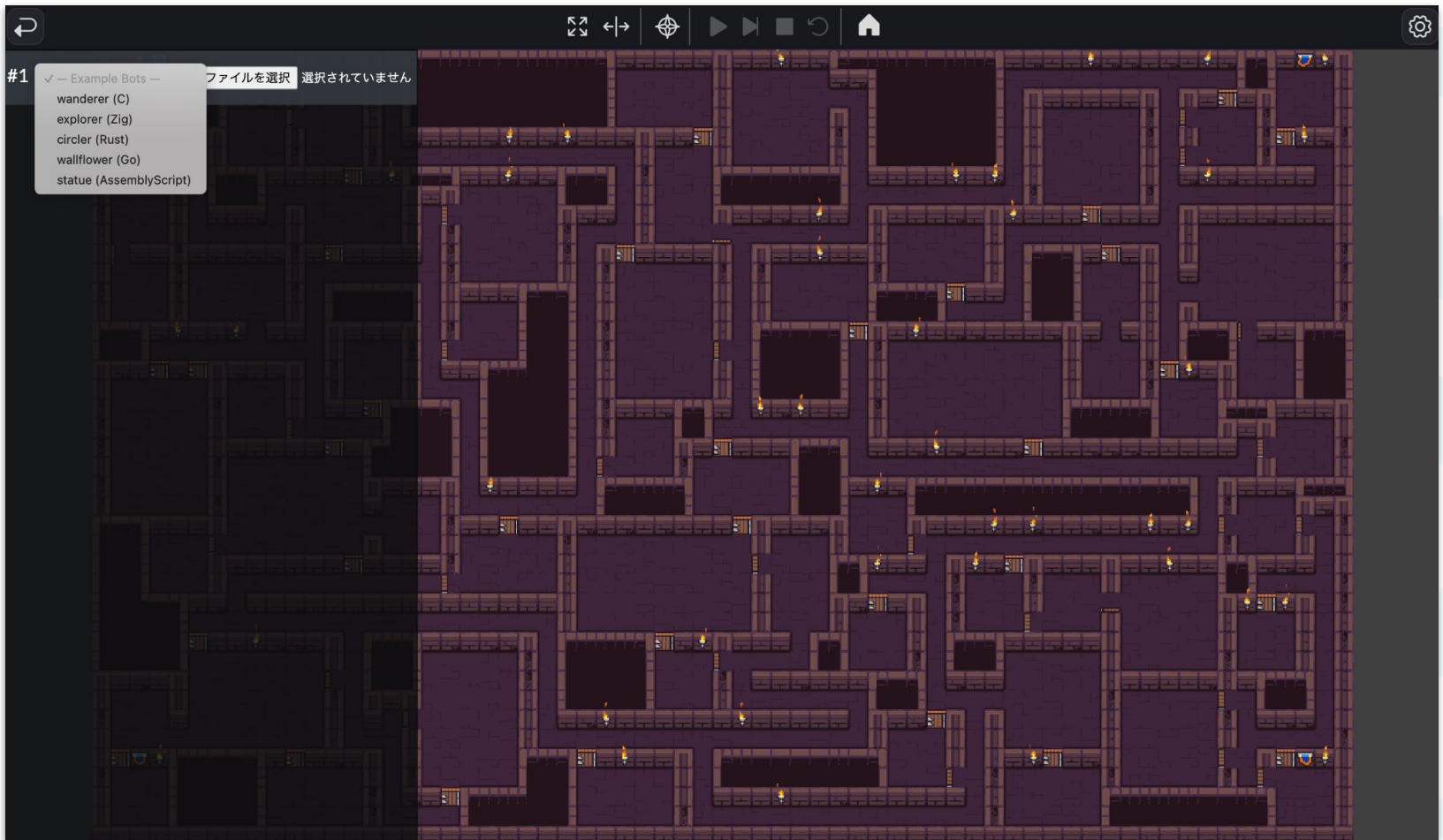
WasmBots: WebAssemblyで動くボットによるマルチプレイヤーゲーム

想像してみてください。ダンジョン探索ゲームで、あなたのコーディングスキルが成功を左右する世界をWasmBotsは、まさにそんな世界を実現する、進行中のプロジェクトです。WebAssembly (Wasm) モジュールを使用して、軽量なダンジョン探索ゲームをプレイすることができますランダムに生成されたダンジョンを、それぞれ異なるプログラミング言語(Zig、Go、C、Rustなど)で書かれた複数のボットが探索します。全てのボットが同じように賢いわけではなく、中には円を描いて走り回っているだけのボットもいるかもしれません。しかしWasmBotsはAIの賢さよりも、システムの機能性について焦点を当てています。

WasmBotsは、Shane Liesegang氏によって開発されています。WasmBotsの開発の歴史については、に記載されている"history.md" ファイルで詳しく知ることができます。WasmBotsの最新ビルドは常にGitHub Pagesで公開されていますが、正式なデプロイ先は別のサイトです。

WasmBotsはマルチプレイヤーゲームであり、全てのプレイヤーはボットとして参加しますプレイヤーは、Wasmモジュールを使用して独自のボットを作成し、ダンジョンに送り込むことができます。ボットは、ダンジョン内を移動したり、アイテムを収集したり、他のボットと戦ったりすることができます。

動いている様子



wasmbotsの対応言語

- ・C/C++
- ・Rust
- ・Zig
- ・AssemblyScript
- ・tinygo、など...



実は、
Rubyでも遊べます

コードです (画面に映し ながら説明)

```
80  ▾ class PresentCircumstances
81  ▾   def initialize(last_tick_duration, last_move_result, hit_points, surroundings)
82      @last_tick_duration = last_tick_duration
83      @last_move_result = last_move_result
84      @hit_points = hit_points
85      @surroundings = surroundings
86  end
87  attr_reader :last_tick_duration, :last_move_result, :hit_points, :surroundings
88  end
89
90
91  def clientInitialize
92      logFunction(LOGLEVEL_INFO, "Hello, world! This is made by #{RUBY_ENGINE}")
93  end
94
95  ▾ def setup(requested_size)
96      logFunction(LOGLEVEL_INFO, "received setup with size: #{requested_size}")
97
98      $memory = SharedMemory.new(requested_size)
99      name = "mruby/edge wasmbot"
100     $memory[0..17] = name
101     # $memory[18..25] = "\0" * 8
102     $memory[26..31] = [0, 2, 0].pack("S S S")
103     $memory
104  end
105
106  ▾ def receiveGameParams(offset)
107      param = $memory[offset..(offset+10)].unpack("S S S C C C")
108      logFunction(LOGLEVEL_INFO, "param version: #{param[0]}")
109      logFunction(LOGLEVEL_INFO, "param engine version: #{param[1]}.#{param[2]}.#{param[3]}")
110      logFunction(LOGLEVEL_INFO, "param diagonal_movement: #{param[4]}")
111      logFunction(LOGLEVEL_INFO, "param player_stride: #{param[5]}")
112      logFunction(LOGLEVEL_INFO, "param player_open_reach: #{param[6]}")
113      true
114  end
115
116  ▾ def tick(offset)
```

本当に動くよ！ demo

まとめ



最近やったことの紹介でした

- ・RustでmrubyのVMを作った
 - ・割とRubyになった
 - ・標準的 or WASIなwasmバイナリを吐き出せる
 - ・色々と使えるはず(プラグイン用WASMとかね)

コントリビュータ募集中です