

はやわかり mruby でばっぐ とらのまき

主要な構造体

mruby_value mruby のオブジェクトを表す

```
typedef struct mrb_value {  
    union {  
        mrb_float f;           浮動小数点数  
        union {  
            void *p;           ポインタ  
            struct {  
                uint32_t ttt;   型のタグ  
                union {  
                    mrb_int i;   整数  
                    mrb_sym sym; シンボル id  
                };  
            };  
        };  
    } value;  
};  
} mrb_value;
```

mruby はオブジェクトは **mruby_value** に型情報と共に入っているので必要に応じてメンバーを参照する。例えば、ポインタは

`v.value.p`

のようにアクセスする。`v` は **mruby_value** 型の変数

(ただし、NaN boxing の場合)

mruby_state mruby の状態を表す構造体。だいたい `mruby` という変数でアクセスできる

メンバーはいっぱいあるのでとりあえずデバッグで使いそうなもの

c context オブジェクト (VM の実行に必要な各種情報 (スタックとか)。**mruby_context** 型へのポインタ

exc 例外があった時例外オブジェクトを格納する

mrb_context VMの実行に必要な情報が入った構造体

メンバーはいっぱいあるのでとりあえずデバッグで使いそうなもの

ci メソッドの呼び出し履歴を格納する

mrb_callinfo 型の配列

stack VM のレジスタ領域(なのに名前は **stack**)

mrb_callinfo メソッドの呼び出し履歴を格納する

メンバーはいっぱいあるのでとりあえずデバッグで使いそうなもの

proc 現在(呼び出し履歴なこの時点での)実行中のメソッドの **Proc** オブジェクト

pc このメソッドが終わった後に実行すべき命令のポインタ

mrb_irep VMの命令列とか定数・シンボルテーブルとかがパックされた構造体

VMはプログラムをだいたい **irep** 単位で扱う。

メンバーはいっぱいあるのでとりあえずデバッグで使いそうなもの

iseq バイトコード列

reps 子 **irep** の配列(Ruby で **lexical** にネストしているブロックの **irep** はここに入る)

RProc **Proc** オブジェクト

とりあえず、**RProc***の変数 **p** があったら、**p->body.irep** とすると **p** の中身の **irep** 構造体が得られる(**Proc** が Ruby で定義されていれば)

主要な命令

面倒なのでここを参照してください

まっとう

http://kimushu.thyme.jp/knowhow/ruby:mruby:instructions#op_move_0x01

ふざけている

<http://qiita.com/miura1729/items/256d205bc2a464bfb3c6>

その他

VMの実行情報は `mruby_vm_exec` のローカル変数として格納されているものが結構あるので、`up/down` 命令を使って `mruby_vm_exec` を定位置にすると吉です

```
p codedump(mrb, irep) または、
```

```
p codedump(mrb, mrb->c->ci->proc.body.irep)
```

とすると現在実行中のメソッドのバイトコード命令列が得られます

これを `.gdbinit` にでも忍ばしておくと `mbt` でバックトレース出来ます

```
define mbt
```

```
  set $p = mrb->c->ci
```

```
  while ($p > mrb->c->cibase)
```

```
    if (($p->proc->flags & 128) != 0)
```

```
      printf "0x%x    C FUNCTION ", $p
```

```
      output $p->proc->body.func
```

```
      printf "¥n"
```

```
    else
```

```
      set $irep = $p->proc->body.irep
```

```
      set $filename = $irep->filename
```

```
      set $lines = $irep->lines
```

```
      if ($filename && $lines)
```

```
        if ($p == mrb->c->ci)
```

```
          set $lineno = $lines[pc - $irep->iseq]
```

```
        else
```

```
          set $lineno = $lines[($p + 1)->pc - $irep->iseq]
```

```
        end
```

```
        printf "0x%x MRUBY FUNCTION : %d %s¥n", $p, $lineno, $filename
```

```
      else
```

```
        printf "0x%x MRUBY FUNCTION : ¥n", $p
```

```
      end
```

```
    end
```

```
    set $p = $p - 1
```

```
  end
```

```
end
```