

Ruby 原生扩展开发指北

@dsh0416



自我介绍

- Midori Developer
- Keio SFC Student
- ~~xdite-buster~~
- MUTA Developer



Ruby 原生扩展开发指北

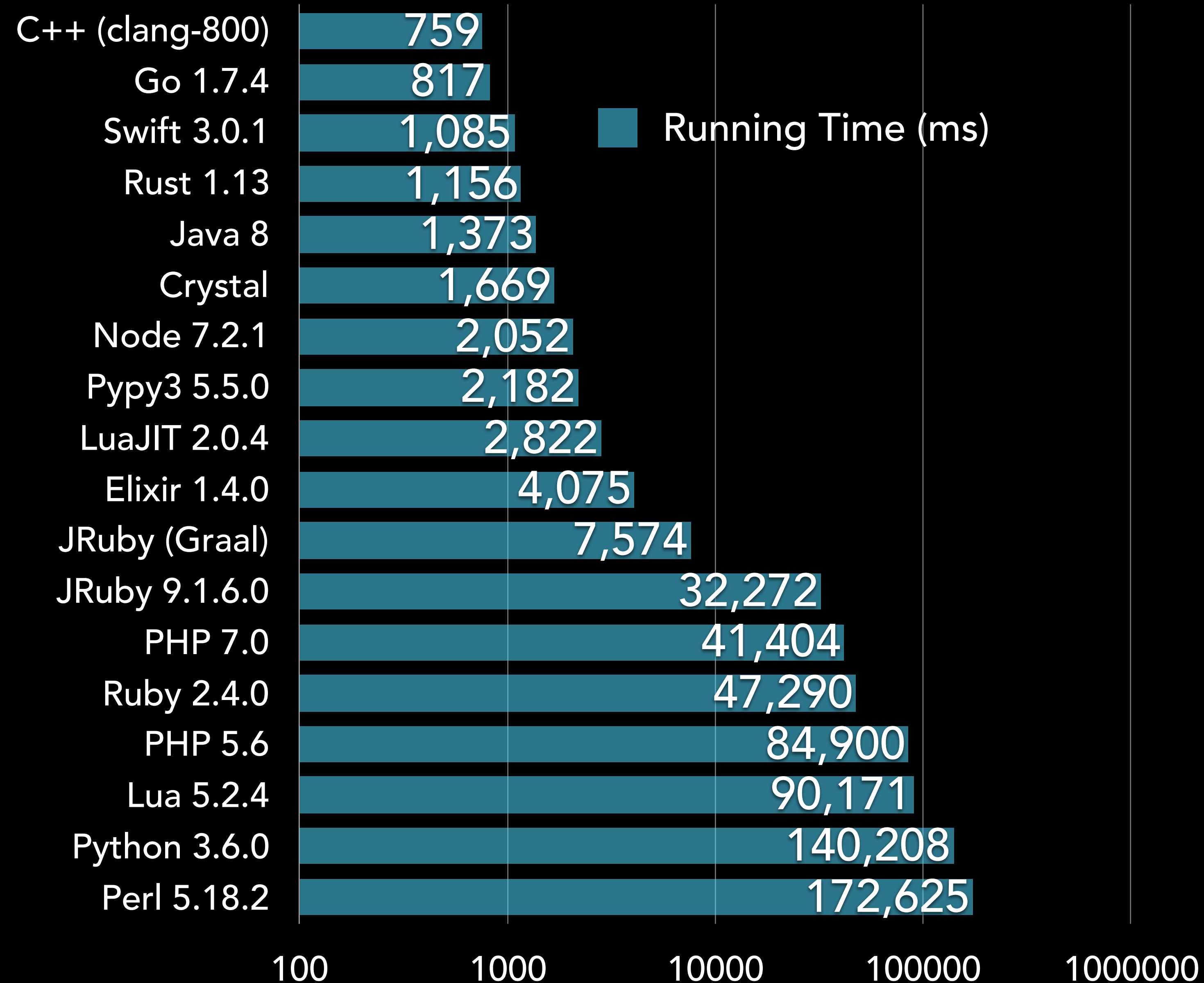
@dsh0416

Native Extensions

Why?

Performance

HEAP SORT (10⁷ ELEMENTS)



Really?

Programmers waste enormous amounts of time thinking about, or worrying about, the speed of noncritical parts of their programs, and these attempts at efficiency actually have a strong negative impact when debugging and maintenance are considered. We should forget about small efficiencies, say about 97% of the time: premature optimization is the root of all evil.

Yet we should not pass up our opportunities in that critical 3%.

– Donald Knuth

- 可能是某个调用极频繁的算法的重新实现
- 可能是某个调用极频繁的字符串处理的重新实现
- 可能是某个操作系统相关的 system call 的组合
- 可能是和某一个其它已有工程的结合
- 而直接用另一门语言写的开销 > 混合开发的开销

Ruby+

How?

C API

FFI

FFI

C

```
typedef struct _WHAT {  
    int d;  
} WHAT;  
  
int doit(WHAT w) {  
    printf("%d\n", w.d);  
    return w.d;  
}
```

Ruby

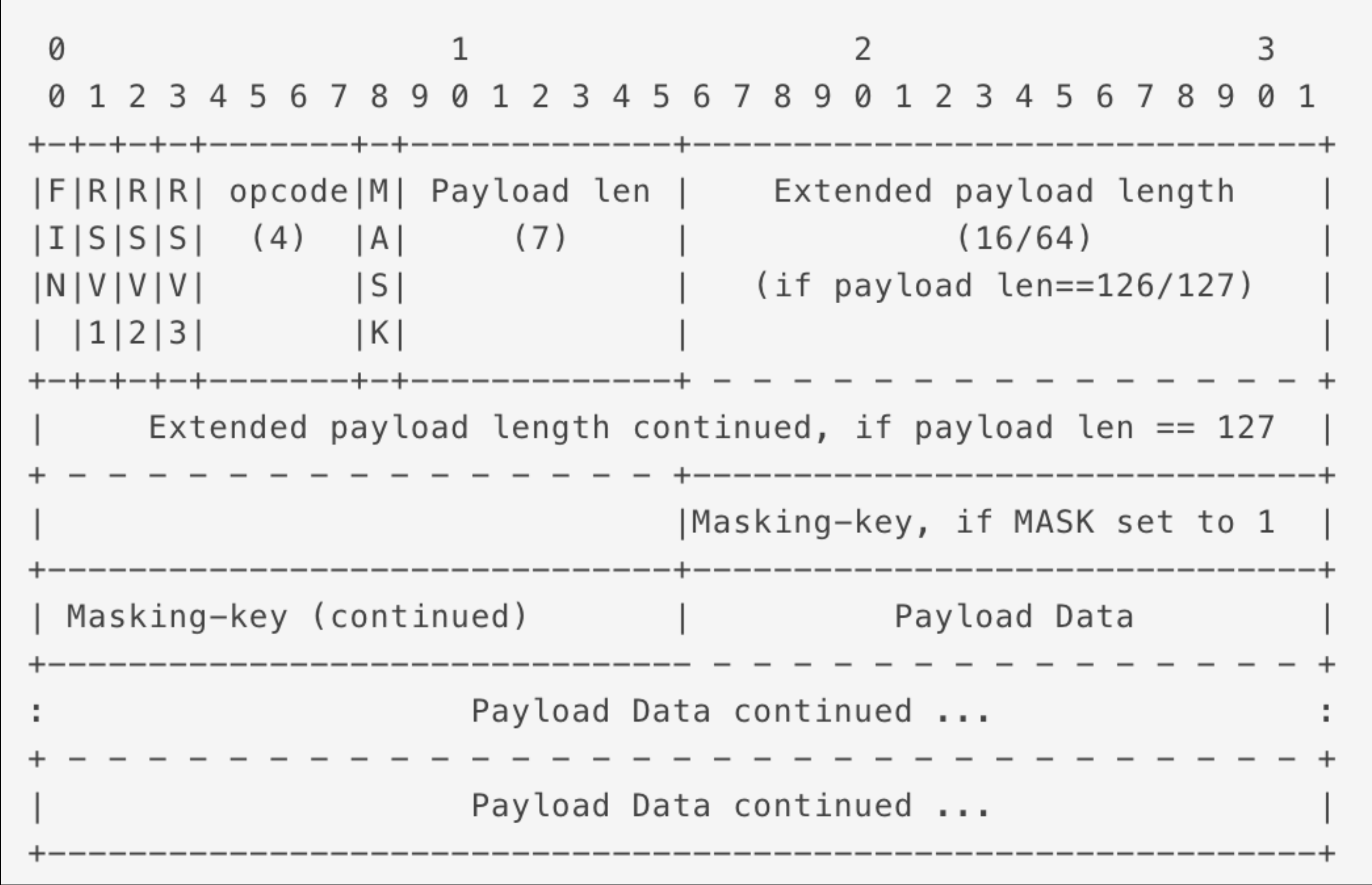
```
class WHAT < FFI::Struct  
    layout :d, :int  
end  
attach_function 'doit', [ WHAT.by_value ], :int
```

缺陷

- 和 Ruby 连接不深，需要手动 attach
- 无法直接操作 Ruby 对象
- 很难和 Ruby 程序做回调操作

C API

直接暴露虚拟机细节使用



0x81 0x85 0x37 0xfa 0x21 0x3d 0x7f 0x9f 0x4d 0x51 0x58

```

#include <ruby.h>
#include <ruby/encoding.h>

VALUE Midori = Qnil;
VALUE MidoriWebSocket = Qnil;

void Init_midori_ext();
VALUE method_midori_websocket_mask(VALUE self, VALUE payload, VALUE mask);
VALUE method_midori_websocket_mask_str(VALUE self, VALUE payload, VALUE mask);
VALUE method_midori_websocket_decode_c(VALUE self, VALUE data);

void Init_midori_ext()
{
    Midori = rb_define_module("Midori");
    MidoriWebSocket = rb_define_class_under(Midori, "WebSocket", rb_cObject);
    rb_define_protected_method(MidoriWebSocket, "mask", method_midori_websocket_mask, 2);
    rb_define_protected_method(MidoriWebSocket, "mask_str", method_midori_websocket_mask_str, 2);
    rb_define_method(MidoriWebSocket, "decode_c", method_midori_websocket_decode_c, 1);
}

VALUE method_midori_websocket_mask(VALUE self, VALUE payload, VALUE mask)
{
    long n = RARRAY_LEN(payload), i, p, m;
    VALUE unmasked = rb_ary_new2(n);

    int mask_array[] = {
        NUM2INT(rb_ary_entry(mask, 0)),
        NUM2INT(rb_ary_entry(mask, 1)),
        NUM2INT(rb_ary_entry(mask, 2)),
        NUM2INT(rb_ary_entry(mask, 3))};

    for (i = 0; i < n; i++)
    {
        p = NUM2INT(rb_ary_entry(payload, i));
        m = mask_array[i % 4];
        rb_ary_store(unmasked, i, INT2NUM(p ^ m));
    }
    return unmasked;
}

```

- 容易做系统调用：epoll、kqueue
- 可以通过调用 C 再调用其它虚拟机的 lib 来跨语言操作

New I/O for Ruby: Cross-platform asynchronous I/O primitives for scalable network clients and servers

📌 585 commits

🌿 2 branches

🏷 35 releases

👤 29 contributors

📄 MIT

Branch: master ▾

New pull request

Create new file

Upload files

Find file

Clone or download ▾

🐱 ioquatix Merge pull request #192 from junaruga/feature/doc-ruby-2.5 ...			Latest commit e84f25e on Jun 11
📁 examples	Add "frozen_string_literal" comments to all files	2 years ago	
📁 ext	Don't preemptively update monitor->interests. Fixes #187.	5 months ago	
📁 lib	Bump version.	5 months ago	
📁 spec	Fix trailing whitespace.	8 months ago	
📁 tasks	Add "frozen_string_literal" comments to all files	2 years ago	
📄 .gitignore	Initial C extension with libev 4.04	7 years ago	
📄 .rspec	Test on Ruby 2.4.0	2 years ago	
📄 .rubocop.yml	Update to RuboCop 0.52.1	10 months ago	
📄 .travis.yml	Update Rubies to the latest version on Travis CI.	4 months ago	
📄 CHANGES.md	v2.3.1	5 months ago	



PyCall: Calling Python functions from the Ruby language

build passing build failing

This library provides the features to directly call and partially interoperate with Python from the Ruby language. You can import arbitrary Python modules into Ruby modules, call Python functions with automatic type conversion from Ruby to Python.

Supported Ruby versions

pypcall.rb supports Ruby version 2.3 or higher.

Supported Python versions

pypcall.rb supports Python version 2.7 or higher.

Note that in Python 2.7 old-style class, that is defined without a super class, is not fully supported in pypcall.rb.

坑

ruby_xcalloc	引数と戻り値は calloc() と同じです。ただしメモリ割り当てに失敗したときは GC を行いそれでもだめなときは例外 NoMemoryError を発生します。つまりこの関数が戻り値を返したときは常に割り当ては成功です。
ruby_xfree	以前 malloc/calloc/realloc して free していないポインタ x を 開放します。ruby のシグナル機構とコンフリクトしません。
ruby_xmalloc	引数と戻り値は malloc() と同じ。ただしメモリ割り当てに失敗したときは GC を行いそれでもだめなときは NoMemoryError を raise する。つまりこの関数が戻り値を返したときは常に割り当ては成功している。
ruby_xrealloc	引数と戻り値の意味は realloc() と同じ。ただしメモリ割り当てに失敗したときは GC を行いそれでもだめなときは NoMemoryError を raise する。つまりこの関数が戻り値を返したときは常に割り当ては成功している。


←

→

×

https://docs.ruby-lang.org/en/2.5.0/

Home

build passing 

Pages Classes Methods

xfree

extract_reference(value)

OpenSSL::Config

What's Ruby

Ruby is the interpreted scripting

日本語を勉強してくれて嬉しい

语言

Why not?

DSL

上午

- 08: 00 – 09: 00 入场
- 09: 00 – 09: 45 Keynote by Matz
- 09: 45 – 10: 30 与 Matz 面对面
- 10: 30 – 11: 15 揭秘码云：基于 Ruby 及 Git 的研发协作平台核心架构 by 码云负责人 周凯
- 11: 15 – 12: 00 Ruby工程师的区块链故事 by HPB CEO 汪晓明

下午

- 13: 15 – 14: 15 Ruby 内存调优 by 侯俊杰（薄荷网）
- 14: 15 – 15: 00 Component Base Rails Application —— 微服务以外的另一种选择 by 陈浩
- 15: 00 – 15: 30 中场休息
- 15: 30 - 16: 15 Ruby 的好伙伴—Rust by 张汉东（《Ruby 原理剖析》译者《Rust编程之道》作者）
- 16: 15 – 17: 00 Ruby 原生扩展开发指北 by 丁盛豪
- 17: 00 - 17: 30 抽奖

晚上

- 18: 00 - 22: 00 晚宴加 Party（感谢码云和 OTCBTC 的特别赞助，地点郑东新区通泰路梧桐树酒店一楼）

```
#[macro_use]
extern crate ruru;

use ruru::{Boolean, Class, Object, RString};

methods!(
    RString,
    itself,

    fn string_is_blank() -> Boolean {
        Boolean::new(itself.to_string().chars().all(|c| c.is_whitespace()))
    }
);

#[no_mangle]
pub extern fn initialize_string() {
    Class::from_existing("String").define(|itself| {
        itself.def("blank?", string_is_blank);
    });
}
```



```
ruby! {  
  class Console {  
    def log(string: String) {  
      println!("LOG: {}", string);  
    }  
  }  
}
```

```
$ rake build  
$ bundle exec irb  
>> require "console"  
>> Console.log("I'm in your Rust")  
LOG: I'm in your Rust  
=> nil
```

缺陷

- 目前开发还属于早期版本
- 当前版本有大量 bug

Thanks

Delton Ding

GitHub: @dsh0416