Ruby on Windows 2010

日本Rubyの会　arton(a.tajima)

Agenda

- 種類
- 特徴
- 問題点
- ASR Way

種類 （Ruby 1.9）

- Cygwin（Perhaps）
- MinGW32（Best effort）
- MSWin32（Best effort）
  - VC6
  - VC7以降
- MSWin64（Best effort）
- MinGW64（Perhaps）
- BCC32（Not supported）

出典：http://redmine.ruby-lang.org/wiki/1/SupportedPlatformsJa

見分け方

- RUBY_PLATFORM 定数を参照する。

```
C:¥tmp>ruby -v
ruby 1.9.1p429 (2010-07-02 revision 28523) [i386-mswin32]

C:¥tmp>ruby -e 'p RUBY_PLATFORM'
"i386-mswin32"
```

- 例）64ビット版か？
  - /mswin64|mingw64/ =~ RUBY_PLATFORM
  - [ruby-dev:41756]

Cygwin

- GCC
- Cygwin

- Ruby 1.4の頃は優勢だったような……

MSWin/MinGWの特徴

- 存在しない標準ライブラリ
  - dbm
    - Win32移植版を用意すれば可能
  - syslog
  - curses
- 意味のない標準ライブラリ
  - etc
    - /etc/passwd、/etc/group
- 使えない関数
  - fork
- 困った記憶はないけど……人によるでしょう

MinGW

- GCC ＋ MSVCRT（VC++6用のlibcのようなもの）

- VC6とGCCの最適化の進化差によって、現在、Windows用Rubyでは最速！！

MSWin32

- Microsoft Visual C++ ＋ MSVCRT

- ≦VC++5
  - 問題外（osfhandleなど）
- VC6 事実上の標準（拡張ライブラリのバイナリ配布）
  - 持っている人だけ
- ≧VC7 バージョンごとに異なるランタイム
  - VC7のライセンス問題
  - VC8の実行権限がらみのいろいろ
  - セキュア強迫症（良いか悪いかはともかく）
  - 無料の開発環境(Visual Studio Express Edition)が入手可能
- 今後は、VC10以降が標準になるかな？

Ruby処理系への影響（１）

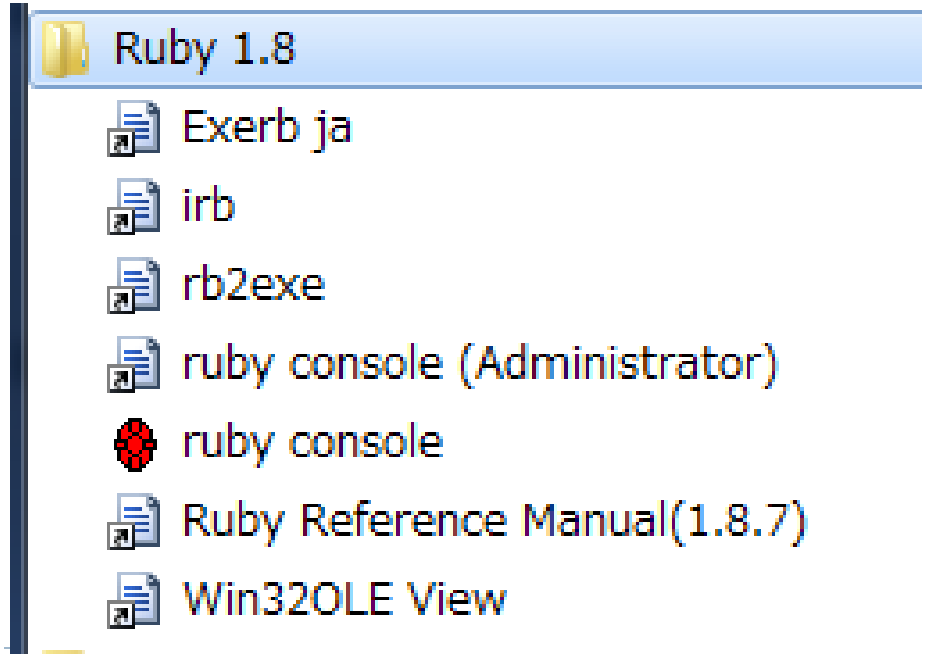
- 無料な環境はフリーな環境
  - Cygwin → 野良ビルド
  - MigGW → 野良ビルド
  - MSWin32 ≧ 7 → 野良ビルド
- 跡地？
  - ftp://ftp.ruby-lang.org/pub/ruby/binaries/

Ruby処理系への影響（２）

- 有償な環境は誰かが作らなければ使えない
- Ruby-mswin32（う）さん提供
  - http://www.garbagecollect.jp/ruby/mswin32/
- One-Click （なんでもあり）
  - http://rubyinstaller.rubyforge.org/wiki/wiki.pl
- ASR （One-Clickではない。元はASR配布目的）
  - http://arton.hp.infoseek.co.jp/indexj.html
  - 標準的な拡張ライブラリの提供（zlib, open-ssl, socket, iconv）
  - 要望があれば入れる（exerb）……たぶん
  - おもしろそうなら入れる（StarRuby、DXRuby）
  - Windowsの作法に準拠
  - Rubyそのものに対するパッチは基本的に入れない

Windowsの作法

- 環境変数への依存を避ける
  - PATHを設定したコンソールの提供
- プログラムは、Program Files / Program Files (x86)
  - short path を使って空白を削除する
- MSIに管理させる
  - MSIパッケージで提供
- UACを生かす
  - 管理者権限を付与するコンソールの提供



Windowsの特徴

- 短所
  - 貧弱なシェル
  - 空白入りディレクトリ名
  - 非POSIX
  - バージョンごとの標準の欠如のように見えるもの
    - （APIを使えば標準フォルダは取れたりする）
  - 16/32ハイブリッドの存在 …… 最近は無視の方向で
- 長所
  - Win32API
  - COM

問題点

- 空白入りディレクトリ
- パスセパレータ、ファイルセパレータ
- ドライブレター

- UAC

- forkの欠如

- 改行コードとBOM（標準エディターがメモ帳……）
  - メモ帳のUTF-8のBOMは多分バグ的なものではないか？（UnicodeのBOMを機械的にUTF-8にエンコードしたのかも。でも、もし自動判別にしくじったらとんでもないことになりそうなので、あえて入れているのかも）

Ruby 1.9

- BOM付きutf-8のスクリプトを実行可能
  - 1.8系はだめ

```
C:¥tmp>c:¥home¥ruby¥bin¥ruby -v bom8.rb
ruby 1.8.7 (2010-06-23 patchlevel 299) [i386-mswin32]
bom8.rb:1: Invalid char `¥357' in expression
bom8.rb:1: Invalid char `¥273' in expression
bom8.rb:1: Invalid char `¥277' in expression
```

  - 自動変換はしない

```
#!/usr/local/bin/ruby -Ku
require 'iconv‘
puts Iconv.conv(‘cp932’, ‘utf-8’, 'こんにちは')
```

- $stdin.getsでもスレッドを実行可能
  - すごいハック！

空白入りディレクトリ（1）

- Rubyの中では使わない。
- PATH環境変数に含めても危険（__FILE__や$0に影響）
- systemの引数をクォートしないプログラム

- Rubyの中では空白を見せなければ良い
  - ShortPathName

空白入りディレクトリ（2）

```
C:¥temp>dir /x
 ドライブ C のボリューム ラベルは OS です
 ボリューム シリアル番号は AEAD-5363 です

 C:¥temp のディレクトリ

2008/10/19  01:16    <DIR>                       .
2008/10/19  01:16    <DIR>                       ..
2008/10/19  01:15    <DIR>          TESTDI~1     test dir
```

空白入りディレクトリ（3）

- Rubyの中では使わない。
- PATH環境変数に含めても危険（起動ディレクトリを参照しておかしくなるものがあってもおかしくない）

- GetShortPathName API
- winpath.rb （ASR同梱）
  - as is copyright
- Pathname#shortname

ASR winpath.rb

```
C:¥temp>ruby -rpathname -e
  'puts(Pathname.glob("test*")[0].realpath)'
C:/temp/test dir

C:¥temp>ruby -rwinpath -e
  'puts(Pathname.glob("test*")[0].shortname)'
C:/temp/TESTDI~1
```

ASR　ruby console

- C:¥Program Filesにインストールしても問題なし
  - （起動されたプログラムはC:¥PROGRA~1¥ruby-1.8¥bin　or lib or share だと考える）

C:¥temp>%PATH%

C:¥PROGRA~1¥ruby-1.8¥bin; c:¥program files¥imagemagick-(略)

C:¥temp>ruby -e p($:)

[“c:/progra~1/ruby-1.8/lib/ruby/site_ruby/1.8”, “c:/progra~1/ruby-1.8/（略）

- ただし、カレントディレクトリがC:¥Program Files¥ruby-1.8¥bin だとダメ（CD優先）

ASR ruby consoleの使い方

- スタートメニューのアイコンをコピー
- 作業ディレクトリへペースト
- プロパティの「作業フォルダ」を空にする。
- またはデスクトップのアイコンの「作業フォルダ」を自分のホームに変える
- 以降は、そのアイコンをエクスプローラでクリック

- カレントディレクトリが異なるため、常にPATHに登録したShortPathNameが使われる
  - 団さんのアイディア

UAC

- ユーザーがシステムファイルを破壊できない仕組み。

- cd /;rm –rf しても大丈夫なように
  cd ¥Windows; rmdir /s * しても大丈夫（たぶん）

- Program Files書き込み禁止
- /usr/binが書き込み禁止と同じこと。
- が、sudoが無い。gemのインストールをどうしろと？

ASR suexec.rb

- ShellExecute API

- suexec.rb
  - fair licence
  - cstructが必要（dl調べるのが面倒だったので）
- SuExec.exec(prog, *args)
  - sudo

- C:¥>ruby -rsuexec -e 'SuExec.exec("notepad.exe")'

ASR　ruby console(Administrator)

- つまり su

- gemの実行
- $: への書き込み、削除など

- 普通に C:¥Windows¥System32¥drivers¥etc とかをいじるのにも便利

VirtualStore問題

- 管理者権限で実行
  - 直接 c:¥Program Files 下を更新
- そうでなければ
  - C:¥users¥___¥AppData¥Local¥VirtualStore を更新
- アンインストール
  - C:¥Program Files下を削除
    - （VirtualStoreは保持したまま）
  - 再インストール後に、古いままのVirtualStoreを参照
- ご利用は計画的に

cstruct

- もともとは、http.sys を使うため
  - freeすべきメモリーブロックをアロケーションして返してくる→どうしろと
- なんか面倒になって中断
- 実装も中途半端
- でも、suexecの役には立っている

cstruct (suexec)

```
ShellExecuteInfoA = C::Struct.define {
   DWORD :cbSize;
   ULONG :fMask;
   HANDLE :hwnd;
   PCSTRA :lpVerb;
   （略）
}

 def self.exec(prog, *params)
   shellExecuteExA(ShellExecuteInfoA.new(
                ShellExecuteInfoA.size,  0, 0,
                  'runas', prog, params.join(' '), '',
                  1, 0, 0, nil, 0, 0, 0, 0).serialize)
```

ASR　おまけライブラリ

- lhalib
  - http://arton.no-ip.info/collabo/backyard/?LhaLib
  - LHa for Unixの単純移植
  - 未だにWindowsだと使っている人がいるし……

lhalib

- http://arton.no-ip.info/collabo/backyard/?LhaLib

```
require ‘lhalib’
LhaLib.x ‘arc.lzh’　#=> arc.lzhの内容を展開

LhaLib.x(‘arc.lzh’) do |info|
  puts(“#{info[:name]} extracted”)
end
```

- Lhaは役目を果たした。過去のアーカイブをxだけできれば良い。という発想からxメソッドだけ実装。

Win32固有のライブラリ

- Win32OLE （助田氏）…… Ruby標準ライブラリ
- SWin（vruby）（nyasu氏）
- Win32API （Moonwolf氏 ？ 1.9ではdlラッパ）
- uconv （よしだむ氏）
- StarRuby （星氏 …… 開発は停止みたい）
- DXRuby（mirichi氏）

- exerb （Yuya氏）

Win32OLE

- RubyからCOMを呼び出す

```
require ‘win32ole’
x = WIN32OLE.new(“com.object”)
x.foo
```

Win32API

- ほとんどのAPIは呼べる（と思う）

```
require 'Win32API'

get_current_process = Win32API.new('kernel32', 'GetCurrentProcess', nil, 'i')
handle = get_current_process.call
is_wow64 = false
begin
  isWow64 = Win32API.new('Kernel32', 'IsWow64Process', ['i', 'p'], 'i')
  bool = "¥0¥0¥0¥0"
  if isWow64.call(handle, bool) != 0
    is_wow64 = bool != "¥0¥0¥0¥0"
  end
rescue RuntimeError
  # no IsWow64Process
end
p is_wow64
```

DXRuby

- RubyからDirectXを呼び出す

```
Window.loop do
  # 左右おした
  angle += Input.x if Input.padPush?(P_LEFT) or Input.padPush?(P_RIGHT)
  angle = 0 if angle > 3
  angle = 3 if angle < 0

  # 3D画面描画
  for i in 0..3
    for j in 0..2
      jx = x + angledata[angle - 3][0] * (j - 1) + angledata[angle- 2][0] * (i - 3)
      iy = y + angledata[angle - 2][1] * (i - 3) + angledata[angle - 3][1] * (j - 1)
      next if iy < 0 or iy > 15 or jx < 0 or jx > 15
      if map[iy][jx] == 1
        Window.draw(0, 0, image[i][j], i - (j - 1).abs)
      end
    end
  end
```

ActiveScriptRuby

- COMクライアントが、IActiveScriptインターフェイスを呼び出す。
- ActiveScriptRubyが、RubyをIActiveScriptサーバとして登録する

```
<script language="RubyScript">
alert("hello world")
</script>
```

COMクライアント……HTA、VB+ScriptControl、WSH……

ASR　サンプル

C:¥Program Files¥ruby-1.8¥samples

biorhythm.hta

　　バイオリズム表示のHTA

ruby.hta

　　あまり意味がないデモ

htaディレクトリ

　　Windows2000マガジンに掲載したHTAなど

rubyize

　　wscを利用してオブジェクトをRuby化

vb

　　VBのフォームをCとV、RubyをMとしたMVC

まとめ

- WindowsでもRubyは使える。それも結構、具合良く

Q&A