

Ruby on Windows

日本Rubyの会 arton(a.tajima)

Agenda

- ▶ 種類
- ▶ 特徴
- ▶ 問題点
- ▶ ASR Way



種類

- ▶ Cygwin
- ▶ MinGW
- ▶ MSWin32
 - ▶ VC6
 - ▶ VC7以降
- ▶ BCC32



Cygwin

- ▶ GCC
- ▶ Cygwin

- ▶ 矛盾した自由(不自由の中の自由)



MinGW

- ▶ GCC + MSVCRT (VC++6用libcのようなもの)
- ▶ 不自由承知の自由というより無料



MSWin32

- ▶ Microsoft Visual C++ + MSVCRT
- ▶ \leq VC++5
 - ▶ 問題外 (osfhandleなど)
- ▶ VC6 事実上の標準
 - ▶ 持っている人だけ
- ▶ \geq 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



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)
 - Windowsの作法に準拠
-



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

▶ 長所

- ▶ Win32API
 - ▶ COM
-



16/32ハイブリッドはなぜだめか

- ▶ 環境変数の容量制限
- ▶ レジストリが中途半端
- ▶ 非同期IO(もともと無理)
- ▶ WinSock2も後付け
- ▶ IEコンポーネントが後付け
- ▶ Unicode APIのサポートが中途半端
- ▶ Win32APIではなくWin32cAPI (cはcommonのc)



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
 - ▶ 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
- ▶ STDIN.getsでもスレッドを実行可能



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るかもしれない)

- ▶ 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` しても大丈夫のように
`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



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;  
  PCSTR :lpVerb;  
  (略)  
}
```

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



ASR おまけライブラリ

▶ Rjb

- ▶ <http://rubyforge.org/projects/rjb/>
- ▶ Ruby-Java Bridge

▶ lhalib

- ▶ <http://arton.no-ip.info/collabo/backyard/?LhaLib>
- ▶ LHa for Unixの単純移植

▶ HeapShow

- ▶ RubyKaigi 2008デモ用



Rjb

▶ <http://rjb.rubyforge.org/svn/>

▶ Ruby Java Bridge

- ▶ OS X
- ▶ Win32
- ▶ Linux

▶ スレッディング周りの制限



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だけでできれば良い。という発想からxメソッドだけ実装。
-



HeapShow

- ▶ <http://svn.arton.no-ip.info/raa/trunk/HeapShow/>
- ▶ 適当にheapポインタテーブルを探しているので、途中で死ぬことが多すぎる。トイプログラムなので真剣に直す気にはならなかったり
- ▶ おもしろいことはおもしろい
- ▶ bitで読んだApple LISPの記事が元ネタ
- ▶ `ruby -rheapshow -e "s=";(1..1000000).each{|i|s<<i.to_s}"`
- ▶ 最後まで走ったら、その日はラッキー
- ▶ CreapShow ロメロ+キングのおばか映画



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 ?)
- ▶ uconv (よしだむ氏)
- ▶ StarRuby (星氏)

- ▶ exerb (Yuya氏)



Win32OLE

- ▶ RubyからCOMを呼び出す

```
require 'win32ole'
```

```
x = WIN32OLE.new("com.object")
```

```
x.foo
```



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.html

動かないデモ

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

▶ ASRを使おう



Q&A

