Ruby master - Bug #9409

Cygwin で "filesystem" の encoding が正しくないケース

01/14/2014 08:43 AM - ganaware (Nayuta Taga)

Status:	Open		
Priority:	Normal		
Assignee:			
Target version:			
ruby -v:	-	**Backport:**	1.9.3: UNKNOWN, 2.0.0: UNKNOWN, 2.1: UNKNOWN

Description

Cygwin で環境変数 LANG に設定されているエンコーディングと
システムのコードページが異なる場合
"filesystem" の encoding が正しく設定されないようです。

例えば、

- Windows 7 (日本語)
- Cygwin 環境 (CYGWIN_NT-6.1-WOW64 ****** 1.7.27(0.271/5/3) 2013-12-09 11:57 i686 Cygwin)
- 環境変数 LANG は ja_JP.UTF-8
- カレントディレクトリに「日本語.txt」という名前のファイルが存在

という状態で以下のコードを実行すると

```
print "LANG=#{ENV['LANG']}\n"
print "\n"
Dir.open('.').each{|item|
  p item.encoding
  p item
}
print "\n"
Dir.open('.',encoding: 'locale').each{|item|
  p item.encoding
  p item
}
print "\n"
```

例えば以下のような出力が得られます。

```
LANG=ja_JP.UTF-8

#<Encoding:Windows-31J>
"."
#<Encoding:Windows-31J>
".."
#<Encoding:Windows-31J>
"test.rb"
#<Encoding:Windows-31J>
"\x{E697}\xA5\x{E69C}\xAC\x{E8AA}\x9E.txt"

#<Encoding:UTF-8>
"."
#<Encoding:UTF-8>
".."
#<Encoding:UTF-8>
"test.rb"
#<Encoding:UTF-8>
"日本語.txt"
```

本来ならば全ての Encoding が UTF-8 であるべきだと思います。

Init_enc_set_filesystem_encoding() を以下のように修正すれば修正可能です。

```
Index: localeinit.c
===================================================================
--- localeinit.c        (revision 44594)
+++ localeinit.c        (working copy)
@@ -53,7 +53,7 @@
     int idx;
 #if defined NO_LOCALE_CHARMAP
 # error NO_LOCALE_CHARMAP defined
-#elif defined _WIN32 || defined __CYGWIN__
+#elif defined _WIN32 && !defined __CYGWIN__
     char cp[sizeof(int) * 8 / 3 + 4];
     snprintf(cp, sizeof cp, "CP%d", AreFileApisANSI() ? GetACP() : GetOEMCP());
     idx = rb_enc_find_index(cp);
```

History

#1 - 01/14/2014 08:45 AM - ganaware (Nayuta Taga)

チケット #9409 が Nayuta Taga によって報告されました。

Bug #9409: Cygwin で "filesystem" の encoding が正しくないケース
https://bugs.ruby-lang.org/issues/9409

- 作成者: Nayuta Taga
- ステータス: Open
- 優先度: Normal
- 担当者:
- カテゴリ:
- 対象バージョン:
- ruby -v: ruby 2.2.0dev (2014-01-14 trunk 44594) [i386-cygwin]
- Backport: 1.9.3: UNKNOWN, 2.0.0: UNKNOWN, 2.1: UNKNOWN ---------- Cygwin で環境変数 LANG に設定されているエンコーディングとシステムのコードページが異なる場合 "filesystem" の encoding が正しく設定されないようです。

例えば、

- Windows 7 (日本語)
- Cygwin 環境 (CYGWIN_NT-6.1-WOW64 ****** 1.7.27(0.271/5/3) 2013-12-09 11:57 i686 Cygwin)
- 環境変数 LANG は ja_JP.UTF-8
- カレントディレクトリに「日本語.txt」という名前のファイルが存在

という状態で以下のコードを実行すると

```
print "LANG=#{ENV['LANG']}\n"
print "\n"
Dir.open('.').each{|item|
p item.encoding
p item
}
print "\n"
Dir.open('.',encoding: 'locale').each{|item|
p item.encoding
p item
}
print "\n"
```

例えば以下のような出力が得られます。

```
LANG=ja_JP.UTF-8

#<Encoding:Windows-31J>
"."
#<Encoding:Windows-31J>
".."
#<Encoding:Windows-31J>
"test.rb"
#<Encoding:Windows-31J>
"\x{E697}\xA5\x{E69C}\xAC\x{E8AA}\x9E.txt"

#<Encoding:UTF-8>
"."
#<Encoding:UTF-8>
".."
```

```
#<Encoding:UTF-8>
"test.rb"
#<Encoding:UTF-8>
"日本語.txt"
```

本来ならば全てのEncodingがUTF-8であるべきだと思います。

Init_enc_set_filesystem_encoding()を以下のように修正すれば修正可能です。

```
Index: localeinit.c
===================================================================
--- localeinit.c	(revision 44594)
+++ localeinit.c	(working copy)
@@ -53,7 +53,7 @@
int idx;
#if defined NO_LOCALE_CHARMAP
# error NO_LOCALE_CHARMAP defined
-#elif defined WIN32 || defined __CYGWIN_
+#elif defined WIN32 && !defined __CYGWIN_
char cp[sizeof(int) * 8 / 3 + 4];
snprintf(cp, sizeof cp, "CP%d", AreFileApisANSI() ? GetACP() : GetOEMCP());
idx = rb_enc_find_index(cp);
```

--
http://bugs.ruby-lang.org/

#2 - 01/14/2014 09:30 AM - nobu (Nobuyoshi Nakada)

- Description updated

filesystem encodingはシステムの設定を反映するので、意図通りの動作ですね。

#3 - 01/14/2014 09:35 AM - nobu (Nobuyoshi Nakada)

- ruby -v changed from ruby 2.2.0dev (2014-01-14 trunk 44594) [i386-cygwin] to -

#4 - 01/14/2014 10:10 AM - ganaware (Nayuta Taga)

Nobuyoshi Nakada wrote:

> filesystem encodingはシステムの設定を反映するので、意図通りの動作ですね。

Windowsネイティブ環境ならばそうかもしれませんが、
Cygwin環境ですのでLANGに設定されているエンコーディングに従うべきではないのでしょうか?

少なくともLANG=ja_JP.UTF-8の状態で
Dir.open('.').each{|item| p item.encoding }
が
#<Encoding:Windows-31J>
になるのはかなり予想外の挙動と言えると思います。

#5 - 01/14/2014 03:29 PM - usa (Usaku NAKAMURA)

Nayuta Taga wrote:

> Windowsネイティブ環境ならばそうかもしれませんが、
> Cygwin環境ですのでLANGに設定されているエンコーディングに従うべきではないのでしょうか?

Cygwinだろうがなんだろうが、Windowsにおいて、ファイルシステムの保持する
エンコーディングをRubyが勝手に違うものとみなすことはできません。

#6 - 01/15/2014 02:01 AM - ganaware (Nayuta Taga)

- File test.c added

Usaku NAKAMURA wrote:

> Nayuta Taga wrote:
>
> > Windowsネイティブ環境ならばそうかもしれませんが、
> > Cygwin環境ですのでLANGに設定されている

エンコーディングに従うべきではないのでしょうか？

> Cygwinだろうがなんだろうが、Windowsにおいて、ファイルシステムの保持する
> エンコーディングをRubyが勝手に違うものとみなすことはできません。

まず、Cygwin 特有の事情をご説明いたします。

http://cygwin.com/cygwin-ug-net/setup-locale.html

> Cygwin uses the setting of the locale environment variables LC_ALL, LC_CTYPE, and LANG, to determine how to convert Windows filenames from their UTF-16 representation to the singlebyte or multibyte character set used by Cygwin.
>
> The setting of the locale environment variables at process startup is effective for Cygwin's internal conversions to and from the Windows UTF-16 object names for the entire lifetime of the current process. Changing the environment variables to another value changes the way filenames are converted in subsequently started child processes, but not within the same process.

つまり Cygwin のファイル名のエンコーディングは、
Cygwin プロセス起動時に設定されていた LANG に左右されます。
(これは一般的なUnixとは異なる挙動だと思います)

例えば、添付ファイル test.c を以下の手順で実行するとします。

1. gcc test.c
2. cygwin プロセスを全て終了させる
3. cmd.exe を起動する
4. c:\> echo test > 日本語.txt
5. c:\> set LANG=ja_JP.SJIS
6. c:\> a.exe > result_sjis.txt
7. c:\> set LANG=ja_JP.UTF-8
8. c:\> a.exe > result_utf8.txt

この結果は以下のようになります。

- result_sjis.txt は SJIS で「日本語.txt」という文字列が含まれる。
- result_utf8.txt は UTF-8 で「日本語.txt」という文字列が含まれる。

ですから、ファイルシステムの保持するエンコーディングはLANG によって決定
されるエンコーディングだとみなしてよいと思われます。

また、コメント1の出力で

```
#<Encoding:Windows-31J>
"\x{E697}\xA5\x{E69C}\xAC\x{E8AA}\x9E.txt"
```

というものがありますが、これは UTF-8 の文字列に誤って Windows-31J とい
うエンコーディングが付加されている状態となります。したがって
item.force_encoding('locale') などとしなければ利用できませんのでかなり
予想外の挙動となります。

Files

test.c	189 Bytes	01/15/2014	ganaware (Nayuta Taga)