Ruby trunk - Feature #11253

rb_io_modestr_oflags for Ruby API

06/12/2015 08:18 AM - naruse (Yui NARUSE)

Status:	Closed	
Priority:	Normal	
Assignee:	matz (Yukihiro Matsumoto)	
Target version:		

Description

If you have a wrapper of IO.open, you may handle mode.
For example on Windows, you may want to add 'b' / OBINARY to given mode.

But some modes has only integer form like O_CLOEXEC, O_EXCL.
If so you need to convert from modestr to oflags

Therefore how about adding a Ruby API version of rb_io_modestr_oflags

Associated revisions

Revision 6a5dda00 - 07/29/2015 01:39 AM - naruse (Yui NARUSE)

- io.c (rb_io_extract_modeenc): add option parameter `flags' to append extra oflags to normal mode. [Feature #11253] [ruby-core:69539]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@51416 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 51416 - 07/29/2015 01:39 AM - naruse (Yui NARUSE)

- io.c (rb_io_extract_modeenc): add option parameter `flags' to append extra oflags to normal mode. [Feature #11253] [ruby-core:69539]

Revision 51416 - 07/29/2015 01:39 AM - naruse (Yui NARUSE)

- io.c (rb_io_extract_modeenc): add option parameter `flags' to append extra oflags to normal mode. [Feature #11253] [ruby-core:69539]

Revision 51416 - 07/29/2015 01:39 AM - naruse (Yui NARUSE)

- io.c (rb_io_extract_modeenc): add option parameter `flags' to append extra oflags to normal mode. [Feature #11253] [ruby-core:69539]

Revision 51416 - 07/29/2015 01:39 AM - naruse (Yui NARUSE)

- io.c (rb_io_extract_modeenc): add option parameter `flags' to append extra oflags to normal mode. [Feature #11253] [ruby-core:69539]

Revision 51416 - 07/29/2015 01:39 AM - naruse (Yui NARUSE)

- io.c (rb_io_extract_modeenc): add option parameter `flags' to append extra oflags to normal mode. [Feature #11253] [ruby-core:69539]

History

#1 - 06/12/2015 03:11 PM - akr (Akira Tanaka)

This issue is discussed at DevelopersMeeting20150612Japan.

We found "flags" keyword argument would be better for the intent.

```
open(filename, "w", flags: File::EXCL)
open(filename, "w", flags: File::SHARE_DELETE)
```

The argument will be bitwise-ORed to oflag for open system call.

If a wrapper of open method want to add flags, it can add flags using integer.

#2 - 07/26/2015 06:11 PM - naruse (Yui NARUSE)

a patch is following:

```
diff --git a/ChangeLog b/ChangeLog
index 23f82a7..83ae426 100644
--- a/ChangeLog
+++ b/ChangeLog
@@ -1,3 +1,9 @@
+Sat Jul 25 22:33:40 2015  NARUSE, Yui  <naruse@ruby-lang.org>
+
+   * io.c (rb_io_extract_modeenc): add option parameter `flags'
+     to append extra oflags to normal mode.
+     [Feature #11253] [ruby-core:69539]
+
 Sat Jul 25 21:03:45 2015  Nobuyoshi Nakada  <nobu@ruby-lang.org>

     * random.c (fill_random_bytes_syscall): get rid of blocking when
diff --git a/NEWS b/NEWS
index 62ddaec..3295c94 100644
--- a/NEWS
+++ b/NEWS
@@ -46,6 +46,10 @@ with all sufficient information, see the ChangeLog file.
     this flag means to permit deleting opened file on Windows, but currently
     this affect only files opened as binary.  [Feature #11218]

+  * new option parameter `flags' is added.
+    this parameter is bitwise-ORed to oflags generated by normal mode argument.
+    [Feature #11253]
+
 * Thread
   * Thread#name, Thread#name= are added to handle thread names [Feature #11251]

diff --git a/io.c b/io.c
index 24fe983..104f521 100644
--- a/io.c
+++ b/io.c
@@ -173,7 +173,7 @@ static VALUE argf;

 #define id_exception idException
 static ID id_write, id_read, id_getc, id_flush, id_readpartial, id_set_encoding;
-static VALUE sym_mode, sym_perm, sym_extenc, sym_intenc, sym_encoding, sym_open_args;
+static VALUE sym_mode, sym_perm, sym_flags, sym_extenc, sym_intenc, sym_encoding, sym_open_args;
 static VALUE sym_textmode, sym_binmode, sym_autoclose;
 static VALUE sym_SET, sym_CUR, sym_END;
 static VALUE sym_wait_readable, sym_wait_writable;
@@ -5340,19 +5340,6 @@ rb_io_extract_modeenc(VALUE *vmode_p, VALUE *vperm_p, VALUE opthash,
     }
     else {
     VALUE v;
-    extract_binmode(opthash, &fmode);
-    if (fmode & FMODE_BINMODE) {
-#ifdef O_BINARY
-            oflags |= O_BINARY;
-#endif
-        if (!has_enc)
-        rb_io_ext_int_to_encs(rb_ascii8bit_encoding(), NULL, &enc, &enc2, fmode);
-    }
-#if DEFAULT_TEXTMODE
-    else if (NIL_P(vmode)) {
-        fmode |= DEFAULT_TEXTMODE;
-    }
-#endif
     if (!has_vmode) {
         v = rb_hash_aref(opthash, sym_mode);
         if (!NIL_P(v)) {
@@ -5364,6 +5351,26 @@ rb_io_extract_modeenc(VALUE *vmode_p, VALUE *vperm_p, VALUE opthash,
         goto vmode_handle;
         }
     }
+    v = rb_hash_aref(opthash, sym_flags);
+    if (!NIL_P(v)) {
+        v = rb_to_int(v);
+        oflags |= NUM2INT(v);
+        vmode = INT2NUM(oflags);
+        fmode = rb_io_oflags_fmode(oflags);
+    }
+    extract_binmode(opthash, &fmode);
+    if (fmode & FMODE_BINMODE) {
```

```
+#ifdef O_BINARY
+            oflags |= O_BINARY;
+#endif
+       if (!has_enc)
+       rb_io_ext_int_to_encs(rb_ascii8bit_encoding(), NULL, &enc, &enc2, fmode);
+   }
+#if DEFAULT_TEXTMODE
+   else if (NIL_P(vmode)) {
+       fmode |= DEFAULT_TEXTMODE;
+   }
+#endif
    v = rb_hash_aref(opthash, sym_perm);
    if (!NIL_P(v)) {
        if (vperm_p) {
@@ -7522,6 +7529,10 @@ rb_io_make_open_file(VALUE obj)
  *  :mode ::
  *    Same as +mode+ parameter
  *
+ *  :flags ::
+ *    Specifies file open flags as integer.
+ *    If +mode+ parameter is given, this parameter will be bitwise-ORed.
+ *
  *  :\external_encoding ::
  *    External encoding for the IO.  "-" is a synonym for the default external
  *    encoding.
@@ -12493,6 +12504,7 @@ Init_IO(void)

     sym_mode = ID2SYM(rb_intern("mode"));
     sym_perm = ID2SYM(rb_intern("perm"));
+    sym_flags = ID2SYM(rb_intern("flags"));
     sym_extenc = ID2SYM(rb_intern("external_encoding"));
     sym_intenc = ID2SYM(rb_intern("internal_encoding"));
     sym_encoding = ID2SYM(rb_intern("encoding"));
diff --git a/test/ruby/test_io.rb b/test/ruby/test_io.rb
index 51b67f3..7adeae4 100644
--- a/test/ruby/test_io.rb
+++ b/test/ruby/test_io.rb
@@ -3230,4 +3230,25 @@ End
     }
   end if /mswin|mingw|bccwin/ !~ RUBY_PLATFORM

+  def test_open_flag
+    make_tempfile do |t|
+      assert_raise(Errno::EEXIST){ open(t, File::WRONLY|File::CREAT, flags: File::EXCL){} }
+      assert_raise(Errno::EEXIST){ open(t, 'w', flags: File::EXCL){} }
+      assert_raise(Errno::EEXIST){ open(t, mode: 'w', flags: File::EXCL){} }
+    end
+  end
+
+  def test_open_flag_binar
+    make_tempfile do |t|
+      open(t, File::RDONLY, flags: File::BINARY) do |f|
+        assert_equal true, f.binmode
+      end
+      open(t, 'r', flags: File::BINARY) do |f|
+        assert_equal true, f.binmode
+      end
+      open(t, mode: 'r', flags: File::BINARY) do |f|
+        assert_equal true, f.binmode
+      end
+    end
+  end if File::BINARY != 0
 end
```

#3 - 07/28/2015 07:35 AM - matz (Yukihiro Matsumoto)

It looks good to me.

Matz.

#4 - 07/29/2015 01:39 AM - naruse (Yui NARUSE)

- Status changed from Assigned to Closed

Applied in changeset r51416.

- io.c (rb_io_extract_modeenc): add option parameter `flags' to append extra oflags to normal mode. [Feature #11253] [ruby-core:69539]