Ruby master - Feature #13869

Filter non directories from Dir.glob

09/05/2017 02:30 PM - naruse (Yui NARUSE)

Status:	Open	
Priority:	Normal	
Assignee:		
Target version:		

Description

Dir.glob is a tool to fetch filesystem entries with filtering.

On Rails, it often query files from template directories with braces (FNM_EXTGLOB)
https://github.com/rails/rails/blob/6f1c18308ebffc97d51440cdeed7be71de58f26a/actionview/lib/action_view/template/resolver.rb#L247

```
      def find_template_paths(query)
        Dir[query].uniq.reject do |filename|
          File.directory?(filename) ||
            # deals with case-insensitive file systems.
            !File.fnmatch(query, filename, File::FNM_EXTGLOB)
        end
      end
```

By this code, File.directory?() will call lstat(2) system call for each files.

But if Dir.glob is extended, it can avoid calling lstat because it can fetch entry's type from struct dirent
(on many platforms).

```
diff --git a/dir.c b/dir.c
index b7afaec4e0..5db9c2dc79 100644
--- a/dir.c
+++ b/dir.c
@@ -206,6 +206,7 @@ typedef enum {
 #else
 #define FNM_SHORTNAME  0
 #endif
+#define FNM_NONDIR 0x40

 #define FNM_NOMATCH    1
 #define FNM_ERROR  2
@@ -1408,7 +1409,7 @@ do_opendir(const int basefd, const char *path, int flags, rb_encoding *enc,
 }

 /* Globing pattern */
-enum glob_pattern_type { PLAIN, ALPHA, MAGICAL, RECURSIVE, MATCH_ALL, MATCH_DIR };
+enum glob_pattern_type { PLAIN, ALPHA, MAGICAL, RECURSIVE, MATCH_ALL, MATCH_DIR, MATCH_NONDIR };

 /* Return nonzero if S has any special globbing chars in it.  */
 static enum glob_pattern_type
@@ -1581,7 +1582,7 @@ glob_make_pattern(const char *p, const char *e, int flags, rb_encoding *enc)
    glob_free_pattern(list);
    return 0;
     }
-    tmp->type = dirsep ? MATCH_DIR : MATCH_ALL;
+    tmp->type = dirsep ? MATCH_DIR : flags & flags & FNM_NONDIR ? MATCH_NONDIR : MATCH_ALL;
     tmp->str = 0;
     *tail = tmp;
     tmp->next = 0;
@@ -1893,7 +1894,7 @@ glob_helper(
     struct stat st;
     int status = 0;
     struct glob_pattern **cur, **new_beg, **new_end;
-    int plain = 0, magical = 0, recursive = 0, match_all = 0, match_dir = 0;
```

```
+    int plain = 0, magical = 0, recursive = 0, match_all = 0, match_dir = 0, match_nondir = 0;
     int escape = !(flags & FNM_NOESCAPE);
     size_t pathlen = baselen + namelen;
     const char *base = path;
@@ -1926,6 +1927,9 @@ glob_helper(
      case MATCH_DIR:
        match_dir = 1;
        break;
+     case MATCH_NONDIR:
+       match_nondir = 1;
+       break;
      case RECURSIVE:
        rb_bug("continuous RECURSIVEs");
    }
@@ -1940,7 +1944,7 @@ glob_helper(
        pathtype = path_noent;
        }
    }
-   if (match_dir && (pathtype == path_unknown || pathtype == path_symlink)) {
+   if ((match_dir || match_nondir) && (pathtype == path_unknown || pathtype == path_symlink)) {
        if (do_stat(fd, base, &st, flags, enc) == 0) {
        pathtype = IFTODT(st.st_mode);
        }
@@ -1953,6 +1957,11 @@ glob_helper(
        status = glob_call_func(funcs->match, subpath, arg, enc);
        if (status) return status;
    }
+   if (match_nondir && pathtype > path_noent && pathtype != path_directory) {
+       const char *subpath = path + baselen + (baselen && path[baselen] == '/');
+       status = glob_call_func(funcs->match, subpath, arg, enc);
+       if (status) return status;
+   }
    if (match_dir && pathtype == path_directory) {
        const char *subpath = path + baselen + (baselen && path[baselen] == '/');
        char *tmp = join_path(subpath, namelen, dirsep, "", 0);
@@ -3152,4 +3161,10 @@ Init_Dir(void)
      *  on Microsoft Windows.
      */
     rb_file_const("FNM_SHORTNAME", INT2FIX(FNM_SHORTNAME));
+
+    /*  Document-const: File::Constants::FNM_NONDIR
+     *
+     *  Makes patterns to match non directory.  Valid only
+     */
+    rb_file_const("FNM_NONDIR", INT2FIX(FNM_NONDIR));
 }
```

History

#1 - 10/19/2017 08:56 AM - knu (Akinori MUSHA)

Just an idea: a flag to say directories should end with a slash so you can collect directories with "...*/" and non-directories with "...*".

#2 - 10/20/2017 01:58 AM - shyouhei (Shyouhei Urabe)

- Description updated

We looked at this issue at yesterday's developer meeting.

"FNM_NONDIR" doesn't sound well (the negative form is problematic). However the attendees could not propose alternative name.

Also there was a discussion as to symbolic links. Should a symlink that points to a directly be included from the results or not? Rails don't want links to directories but that seems not general enough; might depend on use cases. Adding "FNM_NONDIR" might perhaps become a Rails-specific hack. A more generic way, if any, is definitely better.

When we look at other languages for solutions of similar problems, we see Python introduces os.DirEntry structure. That is informative, however might not be good for our situation (new toplevel class is too big to solve this).