Ruby master - Feature #16562
Expose rb_io_set_encoding_internal to reduce function calls on loading source files
01/26/2020 03:56 PM - methodmissing (Lourens Naudé)

Status: Open
Priority: Normal
Assignee:
Target version:

Description
References https://github.com/ruby/ruby/pull/2829

Removes a few rb_funcall calls from source file reading and the AST loader.

- rb_ast_parse_file: removes 1 x rb_funcall, removes 1 transient string allocation (""")
- load_file_internal: removes 3 x rb_funcall, removes 1 transient string allocation (""")
- Removes a branch from rb_io_set_encoding which checks for a T_FILE type which I could not take with ruby tests or Rails
  (asserted with an inline rb_bug for backtrace :smile: ). I declare it dead
- Removes static symbol id_set_encoding from io.c
- Introduces a literal fstring str_no_transcoding to represent the no transcoding option ".".

Now ... I don't think I like the API naming convention of rb_io_set_encoding_internal very much for the following reason: with internal it means not public API (but then again declaring it in ruby/io.h implies that too, but can also be interpreted as internal encoding, which can be confusing.

Open to ideas about the API and also the special Qfalse argument which assigns the no transcoding special case.

Master booting Redmine with Rails 6:

[RUBY_DEBUG_COUNTER] obj_newobj 2095742
[RUBY_DEBUG_COUNTER] frame_push_cfunc 2217390

This PR:

[RUBY_DEBUG_COUNTER] obj_newobj 2093369
[RUBY_DEBUG_COUNTER] frame_push_cfunc 2212686

Diffs:

- 4704 less C function calls (my understanding is rb_call and friends would bump frame_push_cfunc)
- 2373 less transient objects allocated (about half of the dispatch calls also allocated a "." string)

Which is about inline with the loaded features for booting the process:

irb(main):002:0> $LOADED_FEATURES.size
=> 2165

History
#1 - 01/27/2020 01:15 AM - shyouhei (Shyouhei Urabe)

This might be a nitpick but, can you tell us the reason why rb_io_set_encoding_internal has to be exposed (that is, callable from 3rd party extension libraries)? I understand it needs to be non-static but it seems not need to be truly globally visible. I guess I missed something.