When we throw in a require it's incorrectly considered as a raise, making the T_IMEMO throw_data object accessible via rescue (in a contrived case). In regular usage this is unlikely to cause an issue because Module#=== immediately returns false for objects with klass == 0.

With test_throw.rb:

```ruby
throw :extdep, 42
and
class Anything < Exception
  def self.===(_); true; end
end

catch(:extdep) do
  begin
    require './test_throw'
    rescue Anything => e
      p e
  end
end
```

We get

```
in 'p': method inspect' called on unexpected T_IMEMO object (0x00007f5e1486b130 flags=0x10000301a) (NotImplementedError)
```

I've proposed a fix via PR [https://github.com/ruby/ruby/pull/5513](https://github.com/ruby/ruby/pull/5513)

**Description**

Associated revisions

**Revision c79d2e54 - 02/06/2022 02:10 AM - jhawthorn (John Hawthorn)**
Fix TAG_THROW through require [Bug #18562]

Previously this was being incorrectly swapped with TAG_RAISE in the next line. This would end up checking the T_IMEMO throw_data to the exception handling (which calls Module#===). This happened to not break existing tests because Module#=== returned false when klass is NULL.

This commit handles throw from require correctly by jumping to the tag retaining the TAG_THROW state.

**Revision 807dd047 - 02/07/2022 11:08 AM - naruse (Yui NARUSE)**
merge revision(s) c79d2e54748f52c5023b0a1ee441561df9826c17: [Backport #18562]

Previously this was being incorrectly swapped with TAG_RAISE in the next line. This would end up checking the T_IMEMO throw_data to the exception handling (which calls Module#===). This happened to not break existing tests because Module#=== returned false when klass is NULL.
This commit handles throw from require correctly by jumping to the tag retaining the TAG_THROW state.
---
load.c | 2 +-
test/ruby/test_exception.rb | 21 ++++++++++++++++++++++++++++++++++
2 files changed, 22 insertions(+), 1 deletion(-)

Fix TAG_THROW through require [Bug #18562]

Previously this was being incorrectly swapped with TAG_RAISE in the next line. This would end up checking the T_IMEMO throw_data to the exception handling (which calls Module#===). This happened to not break existing tests because Module#=== returned false when klass is NULL.

This commit handles throw from require correctly by jumping to the tag retaining the TAG_THROW state.

#2 - 02/06/2022 10:55 AM - byroot (Jean Boussier)
- Backport changed from 2.6: UNKNOWN, 2.7: UNKNOWN, 3.0: UNKNOWN, 3.1: UNKNOWN to 2.6: DONTNEED, 2.7: DONTNEED, 3.0: DONTNEED, 3.1: DONTNEED

jhawthorn (John Hawthorn) that was a master only issue right? I could repo it with any released version.

#3 - 02/06/2022 06:43 PM - jhawthorn (John Hawthorn)
- Backport changed from 2.6: DONTNEED, 2.7: DONTNEED, 3.0: DONTNEED, 3.1: DONTNEED to 2.6: DONTNEED, 2.7: REQUIRED, 3.0: REQUIRED, 3.1: REQUIRED

byroot (Jean Boussier) wrote in #note-2:

jhawthorn (John Hawthorn) that was a master only issue right? I could repo it with any released version.

Thanks for checking. I didn't realize this before, but it seems that this needs --disable-gems to reproduce, but with that I see the issue on 2.7, 3.0, and 3.1.

$ ruby --disable-gems -e 'puts RUBY_DESCRIPTION; class Anything < Exception; def self.===(_); true; end; end; catch(:extdep) do begin require "/test_throw"; rescue Anything => e; p e; end; end'
ruby 3.1.0p0 (2021-12-25 revision fb4df44d16) [x86_64-linux]
-e:1:in `p': method `inspect' called on unexpected T_IMEMO object (0x00007f3fb90e5a10 flags=0x301a) (NotImplementedError)
from -e:1:in `rescue in <main>'
from -e:1:in 'block in <main>'
from -e:1:in 'catch'
from -e:1:in '<main>'

I added REQUIRED for 2.7 and up, but I think it's unlikely anyone will run into this bug in real usage, so I don't feel strongly that it needs a backport if it's complicated.

#4 - 02/07/2022 11:08 AM - naruse (Yui NARUSE)
- Backport changed from 2.6: DONTNEED, 2.7: REQUIRED, 3.0: REQUIRED, 3.1: REQUIRED to 2.6: DONTNEED, 2.7: REQUIRED, 3.0: REQUIRED, 3.1: DONE

eee441561df9826c17.