The following scripts with autoload causes Ractor to report a deadlock and segfault (tested on Ubuntu 20.04 and macOS 11.5.2):

```ruby
# test.rb
module Foo
  autoload :Bar, "#{File.dirname(__FILE__)}/test2"
  def self.run
    workers = (0...32).map do
      Ractor.new do
        loop do
          Bar.new
          Ractor.yield nil
        end
      end
    end
    100.times { Ractor.select(*workers) }
  end
end
Foo.run

# test2.rb
module Foo
  class Bar
  end
end
```

Deadlock error:

test.rb:8: in `block (3 levels) in run`: No live threads left. Deadlock? (fatal)
1 threads, 1 sleeps current: 0x00007fc84f3f580
main thread: 0x00007fc84f3f580
*<Thread:0x00007fc84f3f580 sleep_forever>
ruby_thread_t: 0x00007fc84f3f580
native: 0x0000700000000 int: 0

from test.rb:7:in `loop'
from test.rb:7:in `block (2 levels) in run'

Segfault:

test.rb:8: [BUG] Segmentation fault at 0x0000000000000005
ruby 3.1.0dev (2021-08-19T08:44:48Z master 6963f8f743) [x86_64-linux]

```
```
Associated revisions
Revision 397a509b - 12/14/2021 05:33 PM - ko1 (Koichi Sasada)
prohibit load by autoload on non-main Ractor
fix [Bug #18120]

History
#1 - 12/02/2021 08:55 PM - ko1 (Koichi Sasada)
- Assignee set to ko1 (Koichi Sasada)
- Status changed from Open to Assigned

#2 - 12/14/2021 10:01 PM - ko1 (Koichi Sasada)
- Status changed from Assigned to Closed

Applied in changeset git|397a509b6d0d1470df8c290d7e4adef78f1532ee.

prohibit load by autoload on non-main Ractor
fix [Bug #18120]