Ruby master - Bug #18484
Fiber should return itself when blocked by IO
01/13/2022 09:57 AM - jakit (Jakit Liang)

Status: Open
Priority: Normal
Assignee: ioquatix (Samuel Williams)
Target version:
ruby -v: ruby 3.1.0p0 (2021-12-25 revision fb4df44d16) [arm64-darwin20]
Backport: 2.6: UNKNOWN, 2.7: UNKNOWN, 3.0: UNKNOWN, 3.1: UNKNOWN

Description
Here's an example:

```ruby
f = Fiber.new do
  sleep(1)
  Fiber.yield 123
end
p f.resume
```

Output:

```
nil
```

Expected output:

```
#<Fiber:0x000000010e825b18@(irb):59 (paused)>
```

In JavaScript, an example like:

```javascript
async function testAsync() {
  return "hello async";
}
const result = testAsync();
console.log(result);
```

Output:

```
Promise { 'hello async' }
```

History
#1 - 01/14/2022 04:37 AM - mame (Yusuke Endoh)
- Assignee set to ioquatix (Samuel Williams)

#2 - 01/14/2022 05:49 AM - ioquatix (Samuel Williams)

```
irb(main):001:1* f = Fiber.new do
  sleep(1)
  Fiber.yield 123
end
irb(main):004:0> end
```

This appears to work fine to me. Can you please give a more detailed example.