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

<table>
<thead>
<tr>
<th>Status:</th>
<th>Open</th>
</tr>
</thead>
<tbody>
<tr>
<td>Priority:</td>
<td>Normal</td>
</tr>
<tr>
<td>Assignee:</td>
<td>ioquatix (Samuel Williams)</td>
</tr>
<tr>
<td>Target version:</td>
<td>ruby 3.1.0p0 (2021-12-25 revision fb4df44d16) [arm64-darwin20]</td>
</tr>
<tr>
<td>Backport:</td>
<td>2.6: UNKNOWN, 2.7: UNKNOWN, 3.0: UNKNOWN, 3.1: UNKNOWN</td>
</tr>
</tbody>
</table>

Description
Here's an example:

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

p f.resume
```

Output:

```ruby
nil
```

Expected output:

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

In JavaScript, an example like:

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

Output:

```javascript
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)

```ruby
irb(main):001:1* f = Fiber.new do
  sleep(1)
  Fiber.yield 123
end
irb(main):004> end
=> #<Fiber:0x0000000010d8aa328 (irb):1 (created)>
irb(main):005> irb(main):006> p f.resume
123
=> 123
irb(main):007>
```

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