Support discovering yield state of individual Fibers

When debugging a multi-Fiber application or library it is useful to be able to look at the stack for the Thread which is possible right now. Currently it is not possible to discover where the Fiber was yielded.

Is this something that could be added?

Thread.backtrace example

---

History

#1 - 04/19/2013 03:49 PM - halorgium (Tim Carey-Smith)
Should this be in common-ruby?
Could someone move it?

#2 - 04/19/2013 04:34 PM - tmm1 (Aman Karmani)
- Project changed from Ruby master to 14

#3 - 04/19/2013 07:55 PM - halorgium (Tim Carey-Smith)
There was some confusion about what I am proposing.
I realised I didn't even explain this!

I would like to be able to see the backtrace of individual Fibers.
This would allow our Celluloid.stack_dump to show each point at which the Fibers are waiting.

#4 - 04/20/2013 08:36 PM - halorgium (Tim Carey-Smith)
I have implemented Fiber#backtrace in Rubinius and have a demo of what I wanted to achieve.

https://gist.github.com/halorgium/c770d3cf27f6279e5e43

This depends on another change, Thread#root_fiber, which returns the root_fiber of the specified Thread.

#5 - 04/28/2013 07:53 AM - ko1 (Koichi Sasada)
- Assignee set to ko1 (Koichi Sasada)

Sorry, I can't understand what is your proposal.
At first example https://gist.github.com/halorgium/f63abf177a96d7113ce3, there is only "thread.backtrace". However, we have already Thread#backtrace.

```
p Thread.current.backtrace
#=>
ruby 2.1.0dev (2013-03-18 trunk 39808) [i386-mswin32_100]
["t.rb:2:in backtrace", "t.rb:2:in"]
```

BTW, please add more specific description.

For example, Rdoc style description for proposed methods is very welcome because we can understand your proposal and we (especially I, who is not good English writer) can copy this document to implemented source code!

#6 - 04/28/2013 07:30 PM - halorgium (Tim Carey-Smith)
When you have many Fibers inside a Thread.
You cannot find where the Fibers are currently suspended.

I propose adding a method to Fiber to return this information.

Calling Fiber.current.backtrace would be equivalent to Thread.current.backtrace due to the nature of Fibers.

Does this help with understanding?

#7 - 02/06/2017 02:58 AM - ko1 (Koichi Sasada)
- Description updated

It seems nice.

#8 - 04/17/2017 05:36 AM - shyouhei (Shyouhei Urabe)
- Status changed from Open to Assigned

#9 - 02/08/2018 09:51 AM - googlefeud (google feud (spammer, locked))
- Subject changed from Support discovering yield state of individual Fibers to google gravity

You can solve your issue by using anti google gravity underwater

#10 - 02/08/2018 10:01 AM - usa (Usaku NAKAMURA)
- Subject changed from google gravity to Support discovering yield state of individual Fibers

#11 - 12/23/2021 11:40 PM - hsbt (Hiroshi SHIBATA)
- Project changed from 14 to Ruby master