Ruby master - Feature #13006

backtrace of thread killer

12/04/2016 05:43 PM - akostadinov (Aleksandar Kostadinov)

Status:	Open	
Priority:	Normal	
Assignee:		
Target version:		

Description

Hello, I am right now having difficulties to understand why one thread I have is dead. Seems been killed by something (rails/puma). But there is no way I can see to understand what and why did it.

That's why I'm proposing a new feature. This would be Thread#kill should record a backtrace at time of call and record it somewhere in the thread object so that later calling Thread#killer_trace would show what performed this call.

Related issues:

Related to Ruby master - Feature #6647: Exceptions raised in threads should b...	**Closed**

History

#1 - 12/06/2016 01:04 AM - shyouhei (Shyouhei Urabe)

- Related to Feature #6647: Exceptions raised in threads should be logged added

#2 - 12/06/2016 01:11 AM - shyouhei (Shyouhei Urabe)

1) Pre-2.4 usage: Killed thread can still be joined. If you call join to a killed thread, that should raise an exception. Its backtrace contains the info you want.

2) Starting from 2.4, in addition to the above, you can set Thread.report_on_exception = true (maybe at the very beginning of your code). This would automatically print backtraces for all killed threads immediately.