Ruby master - Bug #7459
at_exit { Thread.new {} } broken in r37921
11/29/2012 09:19 AM - zenspider (Ryan Davis)

Status: Closed
Priority: Normal
Assignee:
Target version: 2.0.0
ruby -v: 2.0
Backport:

Description
ruby -e 'p Thread.current; at_exit { p Thread.current; [Thread.new []].map(&:join) }'

outputs:
#
#
-e:1:in new': can't alloc thread (ThreadError)
from -e:1:inblock in '

Either the main thread shouldn't be marked as dead until it actually is, or the check in thread_s_new should be expanded to include whatever thread is responsible for the at_exit run.

History
#1 - 11/29/2012 09:40 AM - zenspider (Ryan Davis)
I don't think main thread should be marked dead until after at_exit is done:

% multiruby -e 'p Thread.current; at_exit { p Thread.current }'

VERSION = mri_trunk
CMD     = ~/.multiruby/install/mri_trunk/bin/ruby -e p Thread.current; at_exit { p Thread.current }

# #
RESULT = 0

VERSION = 1.8.7-p249
CMD     = ~/.multiruby/install/1.8.7-p249/bin/ruby -e p Thread.current; at_exit { p Thread.current }

# #
RESULT = 0

VERSION = 1.9.2-p320
CMD     = ~/.multiruby/install/1.9.2-p320/bin/ruby -e p Thread.current; at_exit { p Thread.current }

# #
RESULT = 0

VERSION = 1.9.3-p194
CMD     = ~/.multiruby/install/1.9.3-p194/bin/ruby -e p Thread.current; at_exit { p Thread.current }

# #
RESULT = 0

TOTAL RESULT = 0 failures out of 4

Passed: 1.9.2-p320, 1.8.7-p249, 1.9.3-p194, mri_trunk
Failed:
This is blocking my merge of minitest to trunk for 2.0 preview.

- Status changed from Open to Closed

fixed at r37958