

Ruby - Feature #21156

Prettier RUBY_DEBUG_LOG output in cruby

02/25/2025 09:39 PM - luke-gru (Luke Gruber)

Status:	Open	
Priority:	Normal	
Assignee:		
Target version:		
Description		
Motivation		
The easiest way I find for debugging ractor and thread scheduling issues is to look at the RUBY_DEBUG_LOG output (must be compiled with -DUSE_RUBY_DEBUG_LOG).		
For example, I would run test.rb with miniruby like this:		
RUBY_DEBUG_LOG=stderr RUBY_DEBUG_LOG_FILTER=-vm_lock,ractor,thread make run > debug.txt 2>&1 tee -a debug.txt		
This removes all the vm_lock stuff from the output and we only get ractor and thread scheduling output. However, the output is hard to read. Here's some example output (it's often thousands of lines):		
11: rb_ractor_blocking_threads_dec r->threads.blocking_cnt:1--, r->threads.cnt:1 thread.c:1521 test.rb:51 nt:1 r:#1/3 (1) th:1		
512: blocking_region_end end thread.c:1526 test.rb:51 nt:1 r:#1/3 (1) th:1		
513: rb_mutex_unlock_th 0x0000600003934d40 thread_sync.c:455 test.rb:51 nt:1 r:#1/3 (1) th:1		
514: thread_sched_lock_ th:1 prev_owner:0 thread_pthread.c:1317 <internal:ractor>:711 nt:1 r:#1/3 (1) th:1		
515: thread_sched_deq next_th:0 readyq_cnt:0 thread_pthread.c:732 <internal:ractor>:711 nt:1 r:#1/3 (1) th:1		
516: thread_sched_wakeup_next_thread next_th:0 thread_pthread.c:962 <internal:ractor>:711 nt:1 r:#1/3 (1) th:1		
517: thread_sched_set_running th:1->th:0 thread_pthread.c:694 <internal:ractor>:711 nt:1 r:#1/3 (1) th:1		
518: thread_sched_wakeup_running_thread no waiting threads thread_pthread.c:791 <internal:ractor>:711 nt:1 r:#1/3 (1) th:1 (rth:1,sth:-1)		
519: thread_sched_setup_running_threads +:0 -:1 +ts:0 -ts:0 thread_pthread.c:567 <internal:ractor>:711 nt:1 r:#1/3 (1) th:1 (rth:1,sth:-1)		
520: ractor_sched_lock_ cr:1 prev_owner:0 thread_pthread.c:569 <internal:ractor>:711 nt:1 r:#1/3 (1) th:1 (rth:1,sth:-1)		
521: ractor_sched_unlock_ cr:1 thread_pthread.c:622 <internal:ractor>:711 nt:1 r:#1/3 (0) th:1 (rth:1,sth:-1)		
522: thread_sched_setup_running_threads run:1->0 thread_pthread.c:641 <internal:ractor>:711 nt:1 r:#1/3 (0) th:1 (rth:1,sth:-1)		
523: thread_sched_wait_running_turn th:1 thread_pthread.c:835 <internal:ractor>:711 nt:1 r:#1/3 (0) th:1 (rth:1,sth:-1)		
524: thread_sched_wait_running_turn (nt) sleep th:1 running:0 thread_pthread.c:848 <internal:ractor>:711 nt:1 r:#1/3 (0) th:1 (rth:1,sth:-1)		
525: thread_sched_set_lock_owner th:0 thread_pthread.c:406 <internal:ractor>:711 nt:1 r:#1/3 (0) th:1 (rth:1,sth:-1)		
526: thread_sched_wait_running_turn nt:1 cond:0x00006000008207d0 thread_pthread.c:852 <internal:ractor>:711 nt:1 r:#1/3 (0) th:1 (rth:1,sth:-1)		
and an example of how I would like it to look:		
228: rb_ractor_thread_switch	th:1->1 (same)	nt:1 th:1
229: rb_ractor_blocking_threads_dec	r->threads.blocking_cnt:1--, r->threads.cnt:1	nt:1 th:1
230: blocking_region_end	end	nt:1 th:1
231: rb_thread_io_blocking_call	th:1 fd:6 ev:1	nt:1 th:1
232: rb_ractor_blocking_threads_inc	cr->threads.blocking_cnt:0++	nt:1 th:1
233: blocking_region_begin	thread_id:0x00000002084c0840	nt:1 th:1
234: thread_sched_lock_	th:1 prev_owner:0	nt:1 th:1

```

235: thread_sched_to_waiting_common      dedicated:1                nt:1 th:1
236: native_thread_dedicated_inc        nt:1 1->2                  nt:1 th:1
237: thread_sched_to_waiting_common0    th:1                        nt:1 th:1
238: thread_sched_deq                    next_th:0 readyq_cnt:0     nt:1 th:1
239: thread_sched_wakeup_next_thread    next_th:0                  nt:1 th:1
240: thread_sched_set_running           th:1->th:0                 nt:1 th:1
241: thread_sched_wakeup_running_thread  no waiting threads        nt:1 th:1
(rth:1,sth:-1)
242: thread_sched_setup_running_threads  +:0 -:1 +ts:0 -ts:0       nt:1 th:1
(rth:1,sth:-1)
243: ractor_sched_lock_                 cr:1 prev_owner:0         nt:1 th:1 (
rth:1,sth:-1)
244: ractor_sched_unlock_               cr:1                       nt:1 th:1 (
rth:1,sth:-1)
245: thread_sched_setup_running_threads  run:1->0                   nt:1 th:1
(rth:1,sth:-1)
246: thread_sched_unlock_               th:1                       nt:1 th:1 (
rth:1,sth:-1)
247: unregister_ubf_list                 th:1                       nt:1 th:1 (
rth:1,sth:-1)
248: thread_sched_lock_                 th:1 prev_owner:0         nt:1 th:1 (
rth:1,sth:-1)
249: thread_sched_to_running_common      th:1 dedicated:1         nt:1 th:1 (
rth:1,sth:-1)
250: native_thread_dedicated_dec        nt:1 2->1                  nt:1 th:1 (
rth:1,sth:-1)
251: thread_sched_to_ready_common        th:1 running:0 readyq_cnt:0 nt:1 th:1 (
rth:1,sth:-1)
252: thread_sched_set_running           th:0->th:1                 nt:1 th:1 (
rth:1,sth:-1)

```

I would like to be able to turn off showing the C file location and ruby file location info and for it to be nicely columnized.

Considerations

Before, every different information type was separated by tabs. Maybe some others who work with the RUBY_DEBUG_LOG still like this format, so I would keep it as default and only format it the pretty way with an environment variable like RUBY_DEBUG_LOG_PRETTY_COLUMNS. For hiding file location info I would like to introduce another environment variable RUBY_DEBUG_LOG_NO_FILE_LOCATIONS, and to still have it show these locations by default as it's sometimes useful.

History

#1 - 02/25/2025 09:43 PM - luke-gru (Luke Gruber)

PR here: <https://github.com/ruby/ruby/pull/12809>

For those interested, I also created a simple vim syntax highlighter for ruby debug log output: https://github.com/luke-gru/ruby_debug_log.vim