

- thread.c (thread_create_core, Init_Thread): initialize th->thread_mask.
- vm_core.h (RUBY_VM_INTERRUPTED_ANY): new macro for avoiding bare th->interrupt_flag.
- vm_core.h (RUBY_VM_INTERRUPTED, RUBY_VM_INTERRUPTED): check th->interrupt_mask.
- thread.c (set_unblock_function, rb_thread_schedule): replace th->interrupt_flag with RUBY_VM_INTERRUPTED_ANY()
- signal.c (signal_exec): set up thread->interrupt_mask for preventing recursive trap handler.
- vm_core.h (RUBY_VM_CHECK_INTS, RUBY_VM_CHECK_INTS_BLOCKING): ditto.
- thread.c (rb_threadptr_execute_interrupts): don't process interrupt if it is masked. [Bug #6009] [ruby-core:42524]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@37861 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 37861 - 11/26/2012 10:57 AM - kosaki (Motohiro KOSAKI)

- ruby_atomic.h (ATOMIC_CAS): new macro for compare-and-exchange.
- vm_core.h (struct rb_thread_struct): add interrupt_mask member.
- thread.c (thread_create_core, Init_Thread): initialize th->thread_mask.
- vm_core.h (RUBY_VM_INTERRUPTED_ANY): new macro for avoiding bare th->interrupt_flag.
- vm_core.h (RUBY_VM_INTERRUPTED, RUBY_VM_INTERRUPTED): check th->interrupt_mask.
- thread.c (set_unblock_function, rb_thread_schedule): replace th->interrupt_flag with RUBY_VM_INTERRUPTED_ANY()
- signal.c (signal_exec): set up thread->interrupt_mask for preventing recursive trap handler.
- vm_core.h (RUBY_VM_CHECK_INTS, RUBY_VM_CHECK_INTS_BLOCKING): ditto.
- thread.c (rb_threadptr_execute_interrupts): don't process interrupt if it is masked. [Bug #6009] [ruby-core:42524]

Revision 37861 - 11/26/2012 10:57 AM - kosaki (Motohiro KOSAKI)

- ruby_atomic.h (ATOMIC_CAS): new macro for compare-and-exchange.
- vm_core.h (struct rb_thread_struct): add interrupt_mask member.
- thread.c (thread_create_core, Init_Thread): initialize th->thread_mask.
- vm_core.h (RUBY_VM_INTERRUPTED_ANY): new macro for avoiding bare th->interrupt_flag.
- vm_core.h (RUBY_VM_INTERRUPTED, RUBY_VM_INTERRUPTED): check th->interrupt_mask.
- thread.c (set_unblock_function, rb_thread_schedule): replace

th->interrupt_flag with RUBY_VM_INTERRUPTED_ANY()

- signal.c (signal_exec): set up thread->interrupt_mask for preventing recursive trap handler.
- vm_core.h (RUBY_VM_CHECK_INTS, RUBY_VM_CHECK_INTS_BLOCKING): ditto.
- thread.c (rb_threadptr_execute_interrupts): don't process interrupt if it is masked. [Bug #6009] [ruby-core:42524]

Revision 37861 - 11/26/2012 10:57 AM - kosaki (Motohiro KOSAKI)

- ruby_atomic.h (ATOMIC_CAS): new macro for compare-and-exchange.
- vm_core.h (struct rb_thread_struct): add interrupt_mask member.
- thread.c (thread_create_core, Init_Thread): initialize th->thread_mask.
- vm_core.h (RUBY_VM_INTERRUPTED_ANY): new macro for avoiding bare th->interrupt_flag.
- vm_core.h (RUBY_VM_INTERRUPTED, RUBY_VM_INTERRUPTED): check th->interrupt_mask.
- thread.c (set_unblock_function, rb_thread_schedule): replace th->interrupt_flag with RUBY_VM_INTERRUPTED_ANY()
- signal.c (signal_exec): set up thread->interrupt_mask for preventing recursive trap handler.
- vm_core.h (RUBY_VM_CHECK_INTS, RUBY_VM_CHECK_INTS_BLOCKING): ditto.
- thread.c (rb_threadptr_execute_interrupts): don't process interrupt if it is masked. [Bug #6009] [ruby-core:42524]

Revision 37861 - 11/26/2012 10:57 AM - kosaki (Motohiro KOSAKI)

- ruby_atomic.h (ATOMIC_CAS): new macro for compare-and-exchange.
- vm_core.h (struct rb_thread_struct): add interrupt_mask member.
- thread.c (thread_create_core, Init_Thread): initialize th->thread_mask.
- vm_core.h (RUBY_VM_INTERRUPTED_ANY): new macro for avoiding bare th->interrupt_flag.
- vm_core.h (RUBY_VM_INTERRUPTED, RUBY_VM_INTERRUPTED): check th->interrupt_mask.
- thread.c (set_unblock_function, rb_thread_schedule): replace th->interrupt_flag with RUBY_VM_INTERRUPTED_ANY()
- signal.c (signal_exec): set up thread->interrupt_mask for preventing recursive trap handler.
- vm_core.h (RUBY_VM_CHECK_INTS, RUBY_VM_CHECK_INTS_BLOCKING): ditto.
- thread.c (rb_threadptr_execute_interrupts): don't process interrupt if it is masked. [Bug #6009] [ruby-core:42524]

Revision 37861 - 11/26/2012 10:57 AM - kosaki (Motohiro KOSAKI)

- `ruby_atomic.h` (`ATOMIC_CAS`): new macro for compare-and-exchange.
- `vm_core.h` (`struct rb_thread_struct`): add `interrupt_mask` member.
- `thread.c` (`thread_create_core`, `Init_Thread`): initialize `th->thread_mask`.
- `vm_core.h` (`RUBY_VM_INTERRUPTED_ANY`): new macro for avoiding bare `th->interrupt_flag`.
- `vm_core.h` (`RUBY_VM_INTERRUPTED`, `RUBY_VM_INTERRUPTED`): check `th->interrupt_mask`.
- `thread.c` (`set_unblock_function`, `rb_thread_schedule`): replace `th->interrupt_flag` with `RUBY_VM_INTERRUPTED_ANY()`
- `signal.c` (`signal_exec`): set up `thread->interrupt_mask` for preventing recursive trap handler.
- `vm_core.h` (`RUBY_VM_CHECK_INTS`, `RUBY_VM_CHECK_INTS_BLOCKING`): ditto.
- `thread.c` (`rb_threadptr_execute_interrupts`): don't process interrupt if it is masked.
[Bug #6009] [ruby-core:42524]

Revision 37861 - 11/26/2012 10:57 AM - kosaki (Motohiro KOSAKI)

- `ruby_atomic.h` (`ATOMIC_CAS`): new macro for compare-and-exchange.
- `vm_core.h` (`struct rb_thread_struct`): add `interrupt_mask` member.
- `thread.c` (`thread_create_core`, `Init_Thread`): initialize `th->thread_mask`.
- `vm_core.h` (`RUBY_VM_INTERRUPTED_ANY`): new macro for avoiding bare `th->interrupt_flag`.
- `vm_core.h` (`RUBY_VM_INTERRUPTED`, `RUBY_VM_INTERRUPTED`): check `th->interrupt_mask`.
- `thread.c` (`set_unblock_function`, `rb_thread_schedule`): replace `th->interrupt_flag` with `RUBY_VM_INTERRUPTED_ANY()`
- `signal.c` (`signal_exec`): set up `thread->interrupt_mask` for preventing recursive trap handler.
- `vm_core.h` (`RUBY_VM_CHECK_INTS`, `RUBY_VM_CHECK_INTS_BLOCKING`): ditto.
- `thread.c` (`rb_threadptr_execute_interrupts`): don't process interrupt if it is masked.
[Bug #6009] [ruby-core:42524]

History

#1 - 02/13/2012 09:54 PM - kosaki (Motohiro KOSAKI)

- Status changed from Open to Assigned
- Assignee set to kosaki (Motohiro KOSAKI)

#2 - 11/26/2012 07:57 PM - kosaki (Motohiro KOSAKI)

- Status changed from Assigned to Closed

- % Done changed from 0 to 100

This issue was solved with changeset r37861.

Eric, thank you for reporting this issue.

Your contribution to Ruby is greatly appreciated.

May Ruby be with you.

- `ruby_atomic.h` (ATOMIC_CAS): new macro for compare-and-exchange.
- `vm_core.h` (struct `rb_thread_struct`): add `interrupt_mask` member.
- `thread.c` (`thread_create_core`, `Init_Thread`): initialize `th->thread_mask`.
- `vm_core.h` (RUBY_VM_INTERRUPTED_ANY): new macro for avoiding bare `th->interrupt_flag`.
- `vm_core.h` (RUBY_VM_INTERRUPTED, RUBY_VM_INTERRUPTED): check `th->interrupt_mask`.
- `thread.c` (`set_unblock_function`, `rb_thread_schedule`): replace `th->interrupt_flag` with `RUBY_VM_INTERRUPTED_ANY()`
- `signal.c` (`signal_exec`): set up `thread->interrupt_mask` for preventing recursive trap handler.
- `vm_core.h` (RUBY_VM_CHECK_INTS, RUBY_VM_CHECK_INTS_BLOCKING): ditto.
- `thread.c` (`rb_threadptr_execute_interrupts`): don't process interrupt if it is masked.
[Bug #6009] [ruby-core:42524]