

Ruby - Feature #14487

[PATCH] simplify altstack and enable reuse with thread cache

02/18/2018 09:28 PM - normalperson (Eric Wong)

Status:	Closed	
Priority:	Normal	
Assignee:		
Target version:		
Description		
simplify altstack and enable reuse with thread cache		
Instead of allocating and registering the altstack in different places, do it together to reduce code and improve readability. When thread cache is enabled, storing altstack in <code>rb_thread_t</code> is wasteful and we may reuse altstack in the same pthread.		
This also lets us clearly allow use of <code>xmalloc</code> to allow GC to recover from <code>ENOMEM</code> .		
This seems straightforward and will commit soonish if no objections.		
(Still working on improving + benchmarking thread cache; hardware sucks :<)		

Associated revisions

Revision 475b4aa40b6cf83e57176a063d368d55bf779e7c - 04/20/2018 09:38 PM - Eric Wong

simplify altstack and enable reuse with thread cache

Instead of allocating and registering the altstack in different places, do it together to reduce code and improve readability. When thread cache is enabled, storing altstack in `rb_thread_t` is wasteful and we may reuse altstack in the same pthread.

This also lets us clearly allow use of `xmalloc` to allow GC to recover from `ENOMEM`.

[ruby-core:85621] [Feature #14487]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@63213 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 475b4aa4 - 04/20/2018 09:38 PM - Eric Wong

simplify altstack and enable reuse with thread cache

Instead of allocating and registering the altstack in different places, do it together to reduce code and improve readability. When thread cache is enabled, storing altstack in `rb_thread_t` is wasteful and we may reuse altstack in the same pthread.

This also lets us clearly allow use of `xmalloc` to allow GC to recover from `ENOMEM`.

[ruby-core:85621] [Feature #14487]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@63213 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision c95467e597963a3016ba918235c4400c6e00adb5 - 07/29/2018 10:15 AM - Eric Wong

`thread_pthread.c`: clear altstacks in thread cache at GVL destruction

Otherwise, an altstack may live past `ObjectSpace` destruction and `xfree`-ing the altstack will segfault.

[ruby-core:85621] [Feature #14487]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@64102 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision c95467e597963a3016ba918235c4400c6e00adb5 - 07/29/2018 10:15 AM - Eric Wong

thread_pthread.c: clear altstacks in thread cache at GVL destruction

Otherwise, an altstack may live past ObjectSpace destruction and xfree-ing the altstack will segfault.

[ruby-core:85621] [Feature #14487]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@64102 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision c95467e5 - 07/29/2018 10:15 AM - Eric Wong

thread_pthread.c: clear altstacks in thread cache at GVL destruction

Otherwise, an altstack may live past ObjectSpace destruction and xfree-ing the altstack will segfault.

[ruby-core:85621] [Feature #14487]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@64102 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

History

#1 - 04/20/2018 09:38 PM - normalperson (Eric Wong)

- Status changed from Open to Closed

Applied in changeset trunk|r63213.

simplify altstack and enable reuse with thread cache

Instead of allocating and registering the altstack in different places, do it together to reduce code and improve readability. When thread cache is enabled, storing altstack in rb_thread_t is wasteful and we may reuse altstack in the same pthread.

This also lets us clearly allow use of xmalloc to allow GC to recover from ENOMEM.

[ruby-core:85621] [Feature #14487]

Files

0001-simplify-altstack-and-enable-reuse-with-thread-cache.patch	5.88 KB	02/18/2018	normalperson (Eric Wong)
---	---------	------------	--------------------------