Ruby - Bug #20256

Make required ASAN settings "built in" without the need for ASAN_OPTIONS env var

02/11/2024 10:25 PM - kjtsanaktsidis (KJ Tsanaktsidis)

Status:	Closed		
Priority:	Normal		
Assignee:	kjtsanaktsidis (KJ Tsanaktsidis)		
Target version:			
ruby -v:		**Backport:**	3.0: UNKNOWN, 3.1: UNKNOWN, 3.2: UNKNOWN, 3.3: UNKNOWN

Description

Currently, to compile Ruby head with ASAN, you need to export the env var ASAN_OPTIONS=use_sigaltstack=0:detect_leaks=0. We require use_sigaltstack=0 because Ruby registers signal handlers with its own sigaltstack, and we require detect_leaks=0 because Ruby does not free all of its own memory at exit (unless the RUBY_FREE_AT_EXIT env var is set). It would be good if this could be "built in" to the build system so pepole do not have to remember to do this.

- For the built ruby program itself, we can add a __asan_default_options function; the asan initialization routine will call this to set the asan options for the program.
- We also need to patch mkmf.rb to set detect_leaks=0 on test programs. The test programs get compiled with Ruby's CFLAGS, which will include -fsanitize=address if Ruby is compiled with asan. Then, if any of these test programs are run (like with try_run, which the socket extconf.rb uses), they will exit with failure status for any ASAN errors. Normally test programs don't have any invalid pointers or such, but they often don't bother to free memory at program exit, so we need to turn of asan's leak detection when executing them.

Associated revisions

Revision 1d467f2255112f9e712d5d9aa6f2cd0a102fb56e - 02/11/2024 11:51 PM - KJ Tsanaktsidis

Burn default ASAN options into the built Ruby

- We always need use_sigaltstack=0 because Ruby registers sigaltstack
handlers
- We also need to disable leak detection (unless RUBY_FREE_AT_EXIT is
set - I might experiment later with automatically enabling leak
detection if RUBY_FREE_AT_EXIT is set).

Burning it into the built ruby binary in this way avoids people needing
to remember to start their Ruby program with these flags all the time.

We also need a small fix in mkmf to make sure that test programs also
don't have leak detection enabled (this is never desirable)

[Bug #20256]

Revision 1d467f2255112f9e712d5d9aa6f2cd0a102fb56e - 02/11/2024 11:51 PM - KJ Tsanaktsidis

Burn default ASAN options into the built Ruby

- We always need use_sigaltstack=0 because Ruby registers sigaltstack
handlers
- We also need to disable leak detection (unless RUBY_FREE_AT_EXIT is
set - I might experiment later with automatically enabling leak
detection if RUBY_FREE_AT_EXIT is set).

Burning it into the built ruby binary in this way avoids people needing
to remember to start their Ruby program with these flags all the time.

We also need a small fix in mkmf to make sure that test programs also
don't have leak detection enabled (this is never desirable)

[Bug #20256]

Revision 1d467f22 - 02/11/2024 11:51 PM - KJ Tsanaktsidis

Burn default ASAN options into the built Ruby

- We always need use_sigaltstack=0 because Ruby registers sigaltstack

handlers
- We also need to disable leak detection (unless RUBY_FREE_AT_EXIT is set - I might experiment later with automatically enabling leak detection if RUBY_FREE_AT_EXIT is set).

Burning it into the built ruby binary in this way avoids people needing to remember to start their Ruby program with these flags all the time.

We also need a small fix in mkmf to make sure that test programs also don't have leak detection enabled (this is never desirable)

[Bug #20256]

History

#1 - 02/11/2024 11:51 PM - Anonymous

- Status changed from Open to Closed

Applied in changeset git|1d467f2255112f9e712d5d9aa6f2cd0a102fb56e.

Burn default ASAN options into the built Ruby

- We always need use_sigaltstack=0 because Ruby registers sigaltstack handlers
- We also need to disable leak detection (unless RUBY_FREE_AT_EXIT is set - I might experiment later with automatically enabling leak detection if RUBY_FREE_AT_EXIT is set).

Burning it into the built ruby binary in this way avoids people needing to remember to start their Ruby program with these flags all the time.

We also need a small fix in mkmf to make sure that test programs also don't have leak detection enabled (this is never desirable)

[Bug #20256]