Ruby - Bug #8446

sdbm fails to fetch existing key if many elements in it

05/24/2013 11:22 PM - arton (Akio Tajima)

Status:	Closed		
Priority:	Normal		
Assignee:	hsbt (Hiroshi SHIBATA)		
Target version:			
ruby -v:	ruby 2.1.0dev (2013-05-24 trunk 40916) [x86_64-linux]	**Backport:**	

Description

=begin
When I store many keys in SDBM, it may fail to fetch the value and returns nil.
The maximum number of all readable keys is varied by the value size.
I believe it's best to be able to store and restore as many as disk size is not full, and it's better to raise an exception while storing time if it can't handle the key.

((|Below is the output of the attached test.|))

Run options:

Running tests:

[1/1] TestManyValues#test_many = 0.08 s

1. Failure:
   TestManyValues#test_many [test_sdbm_many.rb:38]:
   <"@@@
   @@@
   @@@
   @@@
   @@@
   @@@
   @@@
   @@@
   @@@
   @@@
   @@@
   @@@
   @@@
   @@@
   @@@
   @@@
   @@@
   @@@
   @1878"> expected but was
   .

Finished tests in 0.089022s, 11.2332 tests/s, 21095.8954 assertions/s.
1 tests, 1878 assertions, 1 failures, 0 errors, 0 skips

=end

History

#1 - 01/30/2014 04:18 AM - hsbt (Hiroshi SHIBATA)

- Target version changed from 2.1.0 to 2.2.0

#2 - 01/05/2018 09:00 PM - naruse (Yui NARUSE)

- Target version deleted (2.2.0)

#3 - 07/13/2019 10:28 PM - jeremyevans0 (Jeremy Evans)

- Backport deleted (1.9.3: UNKNOWN, 2.0.0: UNKNOWN)

This bug still exists. With enough entries in the database, the retrieval error rate gets fairly high (almost 2% at 32k entries):

```
# missed keys, total entries, error rate
2, 4096, 0.00048828125
14, 8192, 0.00170898437
79, 16384, 0.00482177734375
610, 32768, 0.01861572265625
1043, 65536, 0.0159149169921875
1221, 131072, 0.00931549072265625
1291, 262144, 0.004924774169921875
```

From some debugging, this is due to makroom modifying the database, losing access to the key. Before key 4011 is added, a retrieval of key 1878 in getpage has:

```
dbit: 4915
pagb: 608
db->hmask: 4095
```

After key 4011 is added, a retrieval of key 1878 in getpage has:

```
dbit: 134300
pagb: 94816
db->hmask: 131071
```

The data is still in the database, even if it is not reachable. A read for key '1878' attempts to read at offset 97091584 in the pag file, when the correct offset in the pag file would be 29982720. Unfortunately, I don't have more time to look into this.

#4 - 05/20/2020 04:57 PM - jeremyevans0 (Jeremy Evans)

It looks like other people using sdbm are experiencing the same issue:
https://www.reddit.com/r/ruby/comments/gneh4r/anyone_used_sdbm_from_stdlib_anyone_found_it_buggy/

Unless someone has time to debug this issue, I recommend we stop shipping sdbm with Ruby starting with Ruby 3.0. Otherwise people are more likely to try to use it, and may not realize it has issues until they experience data loss. It is already available as a gem for people that want to use it.

#5 - 05/20/2020 10:57 PM - hsbt (Hiroshi SHIBATA)

- Assignee set to hsbt (Hiroshi SHIBATA)

- Status changed from Open to Assigned

I'm +1 to stop to ship with the next stable release.

The users could install from rubygems or http://github.com/ruby/sdbm

#6 - 05/22/2020 01:38 AM - nobu (Nobuyoshi Nakada)

Agreed on removal.
Sdbm seems unmaintained 30 years, and having other bugs (or limitations) on storing large values too.

#7 - 06/18/2020 04:15 AM - mame (Yusuke Endoh)

Agreed.

#8 - 06/18/2020 04:18 AM - matz (Yukihiro Matsumoto)

It was written by me in the early stage of Ruby development as an example extension. I agree to make it separated gem.

Matz.

#9 - 07/28/2020 08:17 PM - jeremyevans0 (Jeremy Evans)

- Status changed from Assigned to Closed

sdbm was removed at 343ad9eff5b8e4c776c1bf193fc125f9ba1cd612.

Files

test_sdbm_many.rb	873 Bytes	05/24/2013	arton (Akio Tajima)