Ruby master - Feature #3608

Enhancing Pathname#each_child to be lazy

07/24/2010 10:27 AM - taw (Tomasz Wegrzanowski)

Status:	Assigned
Priority:	Normal
Assignee:	akr (Akira Tanaka)
Target version:	

Description

=begin
Right now it lists entire directory, then yields
every element, that is x.each_child(&b) means x.children.each(&b).

This is too slow for directories mounted over networked file systems etc.,
and there is currently no way to get lazy behaviour, other than leaving
convenient #each_child/#children API and moving to lower level.

With this patch:

- #children is eager like before, no change here
- #each_child becomes lazy
- #each_child without block returns lazy enumerator, so it can be used like this dir.each_child.find(&:symlink?) without losing laziness.

Patch is against trunk. pathname.rb was in lib/ not ext/pathname/lib/
before, but it works either way.

The part to return enumerator when called without a block wouldn't
work in 1.8. If backport is desired, that line would need to be thrown
away, and #children would need to build result array instead
of calling each_child(with_directory).to_a - this would be straightforward.
=end

History

#1 - 07/29/2010 05:44 PM - akr (Akira Tanaka)

2010/7/24 Tomasz Wegrzanowski redmine@ruby-lang.org:

> Feature #3608: Enhancing Pathname#each_child to be lazy
> http://redmine.ruby-lang.org/issues/show/3608
>
> Right now it lists entire directory, then yields
> every element, that is x.each_child(&b) means x.children.each(&b).
>
> This is too slow for directories mounted over networked file systems etc.,
> and there is currently no way to get lazy behaviour, other than leaving
> convenient #each_child/#children API and moving to lower level.

A problem of the lazy behaviour that is it opens a file descriptor when
the block is called.

If the lazy each_child is used for recursively, the limit of number of
descriptors limits the recursive levels.

I'm not sure which problem is important.

--
Tanaka Akira

#2 - 08/02/2010 05:43 AM - taw (Tomasz Wegrzanowski)

- File lazy_path_test.rb added

A problem of the lazy behaviour that is it opens a file descriptor when the block is called.

If the lazy each_child is used for recursively, the limit of number of descriptors limits the recursive levels.

I'm not sure which problem is important.

This won't normally be a problem as directory handler isn't opened on to_enum, only once iteration actually begins.

Unless you put these enumerators on different fibres or something like that, your maximum number of open files will be limited by your file system depth and also by stack depth, whichever is lower.

You'd need to have 100s of sub directories nested in one another like 1/2/3/4/5/.../100, and have all these nested on ruby stack.

Take a look at attached test code (also at http://pastebin.org/439336)

Even with ulimit -n as low as 16 and a lot of directories it works perfectly (tested on 00/a - 99/z and on ruby source tree).

Test 1 shows that calling map(&:each_child) won't open directory handlers just yet.

Test 2 shows that each_child works all right with recursion.

Test 3 just verifies that ulimit -n is applied.

#3 - 06/26/2011 01:34 PM - akr (Akira Tanaka)

- Project changed from Ruby to Ruby master

- Assignee set to akr (Akira Tanaka)

#4 - 03/18/2012 06:46 PM - shyouhei (Shyouhei Urabe)

- Status changed from Open to Assigned

#5 - 11/20/2012 09:25 PM - mame (Yusuke Endoh)

- Description updated

- Target version set to 2.6

#6 - 12/25/2017 06:14 PM - naruse (Yui NARUSE)

- Target version deleted (2.6)

Files

lazy_each_child.diff	1.19 KB	07/24/2010	taw (Tomasz Wegrzanowski)
lazy_path_test.rb	1.06 KB	08/02/2010	taw (Tomasz Wegrzanowski)