Ruby master - Feature #16425

Add Thread#dig

12/16/2019 12:38 PM - nobu (Nobuyoshi Nakada)

Status:	Open
Priority:	Normal
Assignee:	
Target version:	

Description

Thread has #[] method like as Array, Hash, Struct and so on, but no #dig.

For instance, PP::PPMethods#check_inspect_key in pp.rb can be simplified with the combination of this method and safe navigation operator.

From

```
    def check_inspect_key(id)
      Thread.current[:__recursive_key__] &&
      Thread.current[:__recursive_key__][:inspect] &&
      Thread.current[:__recursive_key__][:inspect].include?(id)
    end
```

To

```
    def check_inspect_key(id)
      Thread.current.dig(:__recursive_key__, :inspect)&.include?(id)
    end
```

Patch: https://github.com/ruby/ruby/pull/2756

History

#1 - 12/17/2019 12:15 AM - shevegen (Robert A. Heiler)

I think the use case has been described and ruby users may agree that this could be an improvement (e. g. such as the example given by nobu for more succinct code).

#dig on Array, Hash and Struct makes sense; for Thread perhaps the net benefit may be a bit more abstract (I guess the most common use case for .dig will be for Array and then possibly Hash ... or vice versa). The more relevant question may then be a language design question, e. g. whether matz thinks that ruby users should/could use dig for Threads too. From this point of view, nobu's comparison makes sense (to me), e. g. the general [] method compared to dig. I myself use [] a lot for custom classes since it is often nice to be able to use both Foobar.new and Foobar[] - the latter even allowing slightly fewer characters. :)

IMO if matz is fine with Thread also being used/usable in that way through .dig then I think nobu's suggestion makes sense, so +1 about the idea.