Ruby master - Feature #16972

Pathname#mkpath to accept block to call on each created directory

06/19/2020 03:16 PM - Dan0042 (Daniel DeLorme)

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

Description

I just had a case where I need to set the permissions on each directory created by Pathname#mkpath, but since this requires to know which directories are created I had to reimplement the mkpath logic like this:

```
file = Pathname.new(Dir.pwd) + "a/b/c/d/e/foo.txt"  #any of these directories may already exist

file.dirname.ascend.take_while{ |d| !d.exist? }.reverse_each do |dir|
  dir.mkdir
  dir.chmod(0775) #rwxrwxr-x
end
```

It occured to me it would be very elegant if mkpath allowed this

```
file.dirname.mkpath{ |dir| dir.chmod(0775) }
```

History

#1 - 06/20/2020 05:15 AM - nobu (Nobuyoshi Nakada)

- Assignee set to akr (Akira Tanaka)

- Status changed from Open to Assigned

- Description updated

Pathname#mkpath is a wrapper of FileUtils.mkdir_p, and the latter has mode: argument to create intermediate directories.
So it would be simple and better to add the same option to Pathname#mkpath for this purpose, I think.
Or do you have any other use cases that this more generic solution is needed?

https://github.com/ruby/ruby/pull/3243

#2 - 06/22/2020 02:55 PM - Dan0042 (Daniel DeLorme)

Thanks for the tip; all these years and I never realized FileUtils.mkdir_p had this mode: argument (and/or never realized what it was for).
I can imagine use cases for this generic solution: chown, chgrp, touch index.html
And in general I think it's better design to have generic/basic building blocks that can be combined in flexible ways.
But I have not personally experienced a need other than chmod, so for my case FileUtils.mkdir_p is enough.