Ruby master - Feature #16913

Add `ARGF#each_io`

05/25/2020 11:33 PM - prajjwal (Prajjwal Singh)

Status:	Open
Priority:	Normal
Assignee:	
Target version:	

Description

Add an iterator for each file supplied on the command line, or STDIN. ARGF#each_io

Current Status

Often, we need to do something with individual files ARGF knows about rather than the concatenation of them. We can combine ARGF#to_io and ARGF#skip to achieve this as follows:

```
while (file = ARGF.to_io)
  break if file.closed? || file.eof?

  csv = CSV.new(file)
  csv.each { |line| p line }

  ARGF.skip
end
```

Proposal

Add an iterator ARGF#each_io to do the above. The above example would then become:

```
ARGF.each_io do |io|
  csv = CSV.new(io)

  csv.each { |line| p line }
end
```

The name is #each_io. We could call it #each_file as well, but ARGF#to_io emits an IO object when the current file is STDIN.

Implementation

A cursory ruby implementation is below. Could better handle the STDIN edge case, and would probably be better off being written in C.

```
def ARGF.each_io(&fn)
  raise 'ARGF#each_io needs a block!' unless block_given?

  while (file = to_io)
    break if file.closed? || file.eof?

    fn.call(file)

    # File was STDIN, no need to go any further.
    break if file.class == IO

    skip
  end
end
```

Issues

- Handling the STDIN edge case is ugly.
- Not clear if eof? checking for STDIN should be left up to the user instead.

- A real world implementation should return a proper iterator instead of the above.