

テキスト処理 第5回 (2006-05-23)

egrep.rb -vのレポート説明

田中哲

産業技術総合研究所

情報技術研究部門

`akr@isc.senshu-u.ac.jp`

<http://staff.aist.go.jp/tanaka-akira/textprocess/>

レポート

- egrep.rb で -v オプションをサポートし、動いている様子を示し、解説せよ
- -v の指定によりマッチ「しない」行を表示する
- -i も同時にサポートすること
- IT's class で 2006-05-23 16:20 まで
- プレインテキスト

egrep.rb

```
require 'optparse'
patopts = 0
opt = OptionParser.new
opt.def_option("-i", "case insensitive") {
  patopts |= Regexp::IGNORECASE
}
opt.parse!
pattern = ARGV.shift
regexp = Regexp.compile(pattern, patopts)
ARGF.each { |line|
  print line if regexp =~ line
}
```

変更方針

```
require 'optparse'
```

```
patopts = 0
```

```
-v オプションを記録する変数を定義する
```

```
opt = OptionParser.new
```

```
opt.def_option("-i", "case insensitive") { ... }
```

```
opt に -v を登録する
```

```
opt.parse!
```

```
pattern = ARGV.shift
```

```
regexp = Regexp.compile(pattern, patopts)
```

```
ARGF.each { |line|
```

```
  print line if regexp =~ line 必要に応じてここを逆にする
```

```
}
```

コード例

```
require 'optparse'
```

```
patopts = 0
```

```
non_match = false
```

```
opt = OptionParser.new
```

```
opt.def_option("-i", "case insensitive") { ... }
```

```
opt.def_option("-v", "select non-matching lines") {  
  non_match = true }  
}
```

```
opt.parse!
```

```
pattern = ARGV.shift
```

```
regexp = Regexp.compile(pattern, patopts)
```

```
ARGF.each { |line|
```

```
  print line if non_match ^ (regexp =~ line)
```

```
}
```

xorで反転させるかどうか制御

いろいろな案

- xor を使わずに if で組み上げる

```
if non_match
  print line if !(regexp =~ line)
else
  print line if regexp =~ line
end
```
- もっと外側で調べる

```
if non_match
  ARGF.each { ... }
else
  ARGF.each { ... }
end
```

典型的な間違い

- -v と -i の組合せを扱っていない
 - v と -i の組合せは 4種類ある
 - どちらも指定されていない
 - -i だけが指定されている
 - -v だけが指定されている
 - -v と -i の両方が指定されている
- `opt.def_option("-v","-i","case insensitive")` とか
`opt.def_option("-v","case insensitive")` だけとか