コマンドラインツール作成ライブラリ Thor による hikiutils の書き換え

情報科学科 西谷研究室 3554 山根 亮太

1 目的

hiki は，hiki 記法を用いた wiki clone である．wiki の特徴である web 上で編集する機能を提供する．研究室内の内部文書，あるいは外部への宣伝資料などに西谷研ではこの hiki system を利用している．初心者にも覚えやすい直感的な操作である．しかし，慣れてくるとテキスト編集や画面更新にいちいち web 画面へ移行せねばならず，編集の思考が停止する．そこで，テキスト編集に優れた editor との連携や，terminal 上の shell command と連携しやすいように hikiutils という CLI(Command Line Interface) を作成して運用している [1]．しかし，そのユーザインタフェースにはコマンドが直感的でないという問題点がある．そこで，optparse[2] というコマンドライン解析ライブラリを使用している hikiutils を，新たなライブラリ Thor[3] を使用してコマンドを書き換え，より直感的なコマンドに変更することが本研究の目的である．

2 optparse による hikiutils の実装

既存の hikiutils はコマンド解析ライブラリの optparse を用いて，コマンドの処理を行っている．optparse は，「コマンドの登録，実行 method」に分けて記述することが期待されている．また，CLI の起動の仕方が独特であり，コマンドの登録と実行が次のように行われる．

- OptionParser オブジェクト opt を生成
- opt にコマンドを登録
- 入力されたコマンドの処理のメソッドへ移動

この様子を元コードから抜粋すると次の通りである．

```
1  def execute
2   @argv << '--help' if @argv.size==0
3   command_parser = OptionParser.new do |opt|
4    opt.on('-v','--version','show␣program␣Version.')
5    { |v|
6      opt.version = HikiUtils::VERSION
7      puts opt.ver
8    }
9    opt.on('-s','--show','show␣sources')
10   {show_sources}
11   ...省略...
12  end
13  ...省略...
14 end
15
16 def show_sources()
17   ...省略...
18 end
19 以下略
```

optparse では OptionParser オブジェクト opt の生成を行い，コマンドを opt に登録することでコマンドを作成することができる．optparse でのコマンドの実行は opt で登録されたコマンドが入力されることでそれぞれのコマンドの処理を行うメソッドに移動し処理を行う．しかし，このコマンド登録は optparse のデフォルトではハイフンを付けたコマンドしか登録ができず，ハイフンなしのコマンド登録はまた別の手段となり，煩雑なコードが必要となる．

3 Thor による実装

新しく適用を考えているコマンド解析ライブラリの Thor(ソーアと発音) は，optparse と同じ機能を提供するが記述の仕方は大きく異なる．Thor での記述例は以下の通りである．

```
1 desc 'show,--show', 'show␣sources'
2 map "--show" => "show"
3 map "-s" => "show"
4 def show
5   ...以下略...
6 end
```

Thor では desc で一覧を表示されるコマンド名，コマンドの説明を登録する．しかし，ここで記述したコマンドは単に一覧で表示させるためのものであり，実際に実行される時に呼び出すコマンド名は，def で定義された名前 (ここでは show) である．Thor では処理実行を行うメソッド名がコマンド名となり，デフォルトではコマンド名 1 つだけが対応する．

これに別名を与えるために利用されるキーワードが map である．map を用いると，B と呼ばれるメソッドを A でも呼べるようにしてくれ，これを使うことでコマンドの別名を指定することができる．これによって，短縮表示 (-s) などを指定するという手段が一般的である．

4 optparse との全体的な比較

Thor と optparse でのコードで最もわかりやすい違いは，Thor のほうがコードが短くなることである．さらに，コマンドの定義も簡単に行うことができ，実行手順も分かりやすくコードが読みやすいため，書き換えもすぐ行うことができた．より直感的にコマンドを実装することが可能となった．

参考文献

[1] https://rubygems.org/gems/hikiutils, 2017/2/16 アクセス.

[2] https://docs.ruby-lang.org/ja/latest/library/optparse.html, 2017/2/16 アクセス.

[3] http://whatisthor.com, 2017/2/16 アクセス.