

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 序論

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。これを便宜上 hiki web system と呼ぶ。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。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 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 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。

2 方法

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 の特徴は、「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 の起動の仕方が特徴的である。optparse では以下の通り、コマンドの登録と実行が行われる。

- 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

```
def execute
  @argv << '--help' if @argv.size==0
  command_parser = OptionParser.new do |opt|
    opt.on('-v','--version','show program Version.')
    { |v|
      opt.version = HikiUtils::VERSION
      puts opt.ver
    }
    opt.on('-s','--show','show sources')
    {show_sources}
    ... 省略...
  end
  ... 省略...
end
```

```
def show_sources()
```

```
... 省略...
```

```
end
```

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る。

3 Thor による実装

```
desc 'show,--show', 'show sources'
```

```
map "--show" => "show"
```

```
def show
```

```
... 以下略...
```

```
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、コマンド名 1 つが対応する。

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 A => B
```

map とは B と呼ばれるメソッドを A でも呼べるようにしてくれるものである。よって、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。

4 optparse との全体的な比較

コードからも Thor のほうが短くなっていることが分かる。よって、Thor と optparse でコードの違いの一部分になるが全体的にも Thor のほうがコードが短くなり、コマンドの定義も簡単に行うことができる。また、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るので、より直感的なコマンドを実装することも可能となった。