

目次

1	卒論の最終形態への変換手順	1
2	hikiutils -i による卒論作成システム	2
3	hikiutils の install	3
4	個別ディレクトリーの構成	3
5	一般的な執筆手順	4
6	rake が用意しているタスク	5
6.1	rake sync	5
6.2	rake convert	6
6.3	rake force_sync	6
6.4	rake chenv	6
6.5	rake increment	6
6.6	rake number	7
7	github による同期	7
8	hiki_help で表示されるヘルプ内容	8
8.1	hiki で卒論を書くときの初期化と掟	8
8.2	図表：すべての図表を keynote にまとめる，タイトルを分かりやすく書く	8

1 卒論の最終形態への変換手順

[[hikiutils_bob_to_latex]] に卒論の最終形態への変換手順が記されています．まずは，下の入門編をお読みください．

2 hikiutils -i による卒論作成システム

卒業論文は，大学で課される一般的なレポートに比べて多くの情報内容が要求される．また，研究という側面から体裁や著作権など多くの出版における掟を遵守しながら，高い品質を保つ必要がある．そのため，指導教官との編集作業が重要となる．編集作業の効率を高めるためには，共同作業を促進するプラットフォームが必要である．

体裁などを考えると Latex の使用が標準であるが，執筆段階での煩わしさを低減するため，Mark Up 記法による文章作成が一般的である（「ドキュメント作成システム構築ガイド，GitHub, RedPen, AsciiDoctor, CI によるモダンライティング」，伊藤敬彦，吉村孝広，技術評論社）．西谷研では Mark Up 記法的一种である hiki 記法を用いて執筆している．hiki 記法から html へ変換するソフト hikidoc による容易な変換が可能のため，日記 web appli の hiki diary でも利用されている．hiki 記法をベースにして，wiki wiki web と同等の機能を提供する hiki システムが github に公開されており，個人の手持ちのパソコンにインストールして wiki を利用することができる．西谷研ではさらに hiki 記法を拡張して，コードのカラー表示，数式の latex 記述が可能となるシステムを利用している．

本資料では，卒論の編集作業を効率化する図に示すようなシステムの使い方を紹介する．

卒論編集システムの概要

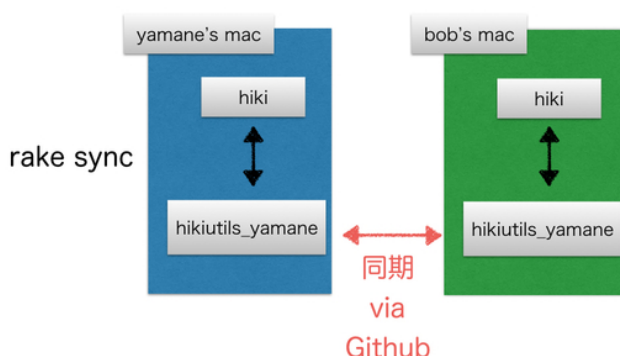


図 1 卒論編集システムの概要．

例えば，卒業生を yamane としよう．yamane の個人の Mac の自分の directory(hikiutils_yamane) に幾つかのファイルを作成して卒論を書いているとする．これを指導教官 (bob) が編集する必要がある．この同期には，Github により提供される共同作業環境をつかう．これだけでは編集集中の文書の体裁がわかりにくい．そこで，hiki システムにより容易に web ブラウザ上に完成形を表示しつつ執筆することが求められる．このような操作環境を提供するのが，hikiutils -i である．

3 hikiutils の install

hikiutils を gem から install しておく必要がある．コマンドは以下の通り．

```
gem install hikiutils
```

さらに

```
hiki -v
```

で 0.2.3.2 以上であることを確認．

4 個別ディレクトリーの構成

図に hikiutils_bob のディレクトリー構成を示す．

コマンド

```
hiki -i
```

によって以下のようなファイルが作成される．

```
bob% ls -lat
total 1072
-rw-r--r--  1 bob  501      99  1 20 12:44 .hikirc
drwxr-xr-x  5 bob  501     170  1 20 12:44 figs/
drwxr-xr-x 12 bob  501     408  1 20 12:34 ./
-rw-r--r--  1 bob  501        8  1 20 11:01 .gitignore
-rw-r--r--  1 bob  501    3507  1 20 11:01 Rakefile
drwxr-xr-x  2 bob  501        68  1 20 11:01 data/
```

個別ディレクトリーの構成

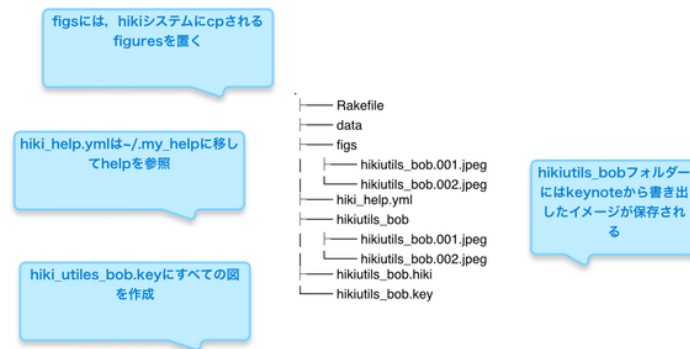


図2 hikiutils.bob のディレクトリー構成 .

```
-rw-r--r--  1 bob  501    2595  1 20 11:01 hiki_help.yml
drwxr-xr-x 26 bob  501     884  1 20 11:00 ../
```

この.hikirc にデータが設定データが自動的に入る．さらに `hikiutils.bob.hiki` および `hikiutils_bob.key` を作成する．これで執筆ファイルの基本構成が出来上がる．

keynote で図を作成して，`hikiutils_bob.hiki` に文章を記述していく．

5 一般的な執筆手順

1. 書類の作成

(a) `open -a mi hikiutils_bob.hiki`

2. keynote を開ける

(a) `open hikiutils_bob.key`

3. keynote のイメージを figs に

(a) keynote でイメージへ書き出し (hikiutils_bob を仮定)

(b) `rake convert 80 hikiutils_bob`

4. hiki システムとの同期

(a) `rake sync`

5. hiki システムで表示

(a) hiki -u hikiutils_bob

6 rake が用意しているタスク

rake の用意しているコマンドは次のとおり .

```
rake check_previous # check previous and current sync
rake chenv          # For hiki Errno::ENOENT, Errno::EACCES
rake convert        # convert fig size SCALE TARGET_DIR
rake force_sync     # force_sync hikis and figs to hiki directory
rake increment      # increment fig NUMBERS in FILE
rake number         # numbering figs from the NUMBER in FILE
rake self_copy      # self copy to hikiutils template directory
rake sync           # normal sync hikis and figs to hiki directory
rake sync0          # sync0 hikis and figs to hiki directory
```

6.1 rake sync

hikiutils_bob にある必要な書類を hiki システムにコピーする . その際 , 名前の書き換えを行う .

表 1

hikiutils_bob での名前	hiki システムでの名前
hikiutils_bob.hiki	hikiutils_bob
introduction.hiki	hikiutils_bob_introduction

figs ディレクトリー内のファイルは hiki/cache/attache/hikiutils_bob に cp される . 従って , hiki 文書中で参照するには , `{{attach_view(hogehoge.png)}}` という記述が必要となる .

6.2 rake convert

keynote が吐き出したイメージを変換するためのコマンド。ImageMagick がインストールされている必要がある。ない場合は、自分で brew から install するか、<https://www.imagemagick.org/script/binary-releases.php> からダウンロード。うまくいかない場合は donkey に聞いてみてください。

```
rake convert 80 hikiutils_bob
```

によって、hikiutils_bob に keynote から吐いた png ファイルを 80

6.3 rake force_sync

hiki システム側で直接変更を加えると、hikiutils が sync した時と差ができる。これを検知して、ユーザに注意を喚起する仕組みがある (rake check_previous)。これは sync した時に自動的に呼び出される。違いの出た files を修正した後に強制的に同期をとるためのコマンドとして、force_sync が用意されている。

6.4 rake chenv

hiki システム上で error が出た場合に試してほしい。error の状況は個人の設定によってちがうため、対処法の実装は網羅されていない。うまくいかない場合は西谷に Issues として投げるように。

6.5 rake increment

keynote でページを追加すると hiki での参照 (attach_view) にずれが生じる。いまのところこれを解消する方法はなく手で修正を加える必要がある。ずれが単純な場合には、

```
cp hikiutils_bob.hiki tmp.hiki
rake increment 2 tmp.hiki > tmp2.hiki
```

として attach_view のページ番号を単純に増加させることができる。

6.6 rake number

前節同じく figs 内の通し番号が変わった時に attach_view の通し番号を調整するコマンド .

```
rake number 3 hikiutils_bob.hiki > tmp.hiki  
cp tmp.hiki hikiutils_bob.hiki
```

とすると

```
8c8  
{{attach_view(hikiutils_bob.002.jpeg)}}  
---  
{{attach_view(hikiutils_bob.003.jpeg)}}  
21c21  
{{attach_view(hikiutils_bob.003.jpeg)}}  
---  
{{attach_view(hikiutils_bob.004.jpeg)}}  
---
```

などと番号を 3 から順に振り替えてくれる .

7 github による同期

1. git init が済んでいると仮定
2. upstream, origin の確認
 - (a) bob
3. git push 作業
 - (a) git add -A
 - (b) git commit -m 'first commit'
 - (c) git push origin master
4. github で bob へ pull request をかける
5. 編集後
6. git pull 作業
 - (a) git pull upstream origin

これがうまくいかん時は聞いてください。

8 hiki_help で表示されるヘルプ内容

hiki_help -i -f により，使い方のメモが表示される。

8.1 hiki で卒論を書くときの初期化と掟

- 開発メモ:figs,data も作成
- 目的：西谷が後で迷わないように決まったファイル構造を堅持すべし
- 文書：hiki で書く。のちには，latex に変換するプログラムを提供します
- 図表：すべての図表を keynote にまとめる，タイトルを分かりやすく書く
- データ：data ディレクトリにまとめる．ファイル名を keynote の対応する図表中に記す
- hiki -initialize で初期ファイル (Rakefile, ./hikirc, hiki_help.yml) が copy される
- hiki_help.yml を適宜 ./my_help に copy して hiki_help として利用，(my_help 参照)
- Errno::EACCES や permission error がでたときは rake chenv を試してみる（報告して）
- rake sync によって hiki ディレクトリーと同期が取られる
- hiki -u TARGET によってブラウザー表示される
- テキストの拡張子は'.hiki'
- hiki での url はテキスト前とディレクトリーから自動生成される
- 例えば，hiki2latex_saki/introducton.hiki とすると hiki2latex_saki.introducton と変換される

8.2 図表：すべての図表を keynote にまとめる，タイトルを分かりやすく書く

- keynote に書いたスライドはイメージに書き出して，rake convert 80 TARGET_DIR で figs に変換
- rake sync で figs にある files は hiki/target_dir に cp される

convert #{source} -resize 20% #{target}によって ,target=figs/TAERGET.png

に 20% に縮小して保存される

- `convert -density 300 view.svg view.png` で 300dpi で変換
- `attach_anchor` では

`'{{attach_anchor(test.png, hiki2latex_saki)}}'` と , `directory` 指定しなければならない.

- `keynote` であとで図を挿入して番号が変わった時の原稿の一括変換
- `rake increment 2 boundary_bob.hiki boundary_bob ; tmp.hiki`
- `rake convert 60 boundary_bob`
- `rake sync`
- `hiki -u boundary_bob_tmp`