

Contents

1	実際の最終形態 (卒論=pdf) への変換	1
1.1	install	1
1.2	rake latex (個別ファイルの変換)	1
1.2.1	注意	2
1.3	rake latex_all 手順 (ディレクトリー内の一括変換)	2
1.3.1	下準備	2
1.4	補助コマンドの解説	3
1.4.1	wrap 関係	3
1.4.2	rake reset_latex_dir(latex_dir のゴミ掃除)	4

1 実際の最終形態 (卒論=pdf) への変換

hiki+keynote で卒論の内容ができたなら、それを卒論の最終形態つまり pdf へ変換する必要があります。

ここで問題が発生します。hiki よりも latex の方が機能が豊富なこと、つまり、簡易に書くには hiki などの mark down で書いていくのがいいんですが、文書として完成度を高めるには、latex で細かい設定を調整する必要があると出てきます。

具体的には、

1. 図の配置を調整する wrap の数値調整
2. 参考文献の記述
3. リスト，引用の体裁
4. 章立て階層構造

などが問題になるところです。

これは hiki ではどうしようもありません。一つの手はもう少し高機能な mark up 言語例えば asciidoc などに変更することですが、これはどこまでいっても終わりのない方向のようで、結局は latex で書いているのと同じになる可能性があります。

我々は違う戦略をとります。それは、

latex をベースにして、hiki を生成する

と言う手です。

DRY(Don't Repeat Yourself) 原則さえ守れば、文書管理はいいのですから、ある段階までおすすめは hiki ではなく latex を原本にするのです。そのための変換器 latex2hiki とその派生 rake 環境が用意されています。そちらはこちらで卒業後に変換します。参考文献は latex に移してから修正してください。

ここでは、hiki でできることをとことん突き詰めておきます。

1.1 install

```
hiki -i
```

で install されています。新たに使うコマンド群は次の通りです。

```
rake latex          # latex conversion FILE1
rake latex_all      # latex conversion whole hiki files in the current dir
rake change_wrap    # change latex figures to wrap format
rake latex_base     # latex conversion FILE1(hiki) to FILE2(latex)
rake latex_wrap     # latex conversion FILE1 with wrap format
rake reset_latex_dir # reset latex dir
```

1.2 rake latex (個別ファイルの変換)

```
rake latex sync.hiki
```

とすると、

```
latex_dir/sync.tex
```

に latex 変換後のファイルが生成します。これを、TeXShop で command.t 変換します。完成例はこちらです。

- `{\attach_anchor(sync.pdf,hikiutils_bob)}`

うまくいかないときは、terminal で

```
platex sync.tex
dvipdfmx sync.dvi
```

を試してみてください。

1.2.1 注意

hiki の初めの部分は、

```
bob% head -3 hikiutils_bob.hiki
! title:hikiutils -i による卒論作成システム %:
! autor:Shigeto R. Nishitani %:
! date: Kwansei Gakuen Univ., 2017/1 %:
```

とすると、

```
\begin{document}
\title{hikiutils -i による卒論作成システム}
\author{Shigeto R. Nishitani}
\date{ Kwansei Gakuen Univ., 2017/1}
\maketitle
```

と変換してくれます。次節の latex_all では、basename.hiki に書かれたそれらの情報は、head.tex で title などを用意するべきなので、自動で消されます。

必要な図は、figs から自動で取るようになっていますが、サイズや解像度が問題のときは手動で調整してください。

1.3 rake latex_all 手順 (ディレクトリー内の一括変換)

pwd の directory と同名の basename に.hiki の拡張子がついたファイルが用意されていて、

```
rake latex_all
```

をおこなうと自動で全部を一体にまとめた文書への latex 変換が出来上がります。たとえば、hikiutils_bob.hiki の記述を

```
bob% cat hikiutils_bob.hiki
!hikiutils を用いた卒業論文作成

! [[hikiutils_bob_sync]]
! [[hikiutils_bob_latex_all]]
```

とすると、sync.tex, latex_all.tex が latex_dir 内に変換されます。さらに、開いている hikiutils_bob.tex を TeXShop で変換してみてください。

うまくできないときはここ (http://qiita.com/hideaki_polisci/items/3afd204449c6cdd995c9) を参照して自力で入れてみてください。だめなら donkey。

- `{\attach_anchor(hikiutils_bob.pdf,hikiutils_bob)}`

というようになります。

1.3.1 下準備

latex_dir 内に幾つかの tex 雛形を入れておく必要があります。

```
hiki -i
```

と hikiutils 環境を再度初期化すると、自動でインストールする設定です。なかったら手動で作ってください。Rakefile 以外は上書きしない。latex_dir/head.tex は修正が必要。

head.tex 題目, 学生番号, 氏名を変更する. 年月をチェック.
setcountertocdepth は toc をどこまで表示するかレベルに対応します. 原稿作成時は階層がわかりやすいように深めにするが, 本番では 2 程度で十分.

```
bob% cat head.tex
\title{卒業論文\\
\vspace{4cm} hikiutils を用いた\\卒業論文作成}
\author{ 関西学院大学 理工学部 情報科学科\\\1234 西谷滋人}
\date{\vspace{3cm} 2017 年 3 月\\
\vspace{3cm} 指導教員 西谷 滋人 教授}
\maketitle
\setcounter{tocdepth}{6} %
\tableofcontents
```

pre.tex あまり変更しないほうがいいが, いずれいじることになります. 例えば, フォントポイント数を 10 から 12pt に変える, を外すなどです.

```
bob% cat pre.tex
\documentclass[10pt,a4j]{article}

\def\Vec#1{\mbox{\boldmath $#1$}}
\usepackage[dvipdfmx]{graphicx}

\setlength{\textheight}{275mm}
\headheight 5mm
\topmargin -20mm
\textwidth 160mm
\textheight 250mm
\oddsidemargin -0mm
\evensidemargin -5mm

\pagestyle{empty}
\makeatletter
\def\@maketitle{%
\newpage\null
\vskip 2em%
\begin{center}%
\let\footnote\thanks
{\large\bf \@title \par}%
\vskip 1.5em%
{\large\bf \@author \par}%
\vskip 1.5em%
{\small \@date}%
\end{center}%
}
\makeatother

%\documentclass[10pt, a4j]{article}
%%\usepackage{citesort}
\usepackage{amssymb}
\usepackage[dvipdfmx]{graphicx}% 図を入れるときに使用
\usepackage{wrapfig}% 図の周りに本文を流し込みたいときに使用
\usepackage{subfigure}
\usepackage{here}
```

1.4 補助コマンドの解説

1.4.1 wrap 関係

figure 環境を wrapfigure 環境で作るための幾つかのコマンド群. 卒論では figure 環境で作る方がいいが, journal 論文などのページ数が制限された場合は, wrapfigure で text の回りこみや位置調整を行う必要があ

る. それらのための環境を埋め込む.

- rake change_wrap(wrap で変換)
- rake latex_base(latex に変換するだけの下請け)
- rake latex_wrap(figure 環境だけを wrapfig 環境に変える)

```
desc "latex conversion FILE1"
task :latex => [:latex_base] do
  exit
end

desc "latex conversion FILE1 with wrap format"
task :latex_wrap => [:latex_base, :change_wrap] do
  exit
end
```

完成例はこちらです.

- `{{attach_anchor(calphad_bob.pdf,hikiutils_bob)}}`

1.4.2 rake reset_latex_dir(latex_dir のゴミ掃除)

わかりやすいようにまとめたり, ファイルの名称を変更した時には, 過去の tex ファイルが残る. それらを整理する時に使用. head.tex だけを escape して rm -rf で消すので, 注意が必要. 特に図形の files や bb files は消えるので注意.