

kagiPDF

崎間

2006年1月26日

kagiPDF は、reStructuredText^{*1}（物理のかぎプロジェクト^{*2}での拡張命令を含む）から、「 \LaTeX によるいい感じの日本語 PDF ファイル」をできるだけ簡単につくるためのコマンドラインツールです。

1 動作環境

Linux や Cygwin などの環境で動作します。動作確認は Vine Linux 3.2 および Windows XP 上の Cygwin 1.5.13 にて行っています。実行にはつぎのソフトウェアが必要です。特に、Docutils と lv は最初から入っていない場合が多いですから、配布元から入手してインストール^{*3}しておいてください。

- Docutils 3.7 or later (<http://docutils.sourceforge.net/>)
- lv (<http://www.ff.iij4u.or.jp/~nrt/lv/>)
- Ruby 1.8 or later
- pLaTeX
- jsarticle, jsbook（日本語 LaTeX ドキュメントクラス）
- dvipdfmx
- ImageMagick

2 インストール

2.1 Linux

bin ディレクトリにある 3 つのファイル extension.rb, fairlatex.rb, kagipdf.rb をパスの通ったディレクトリに置いてください（3 つとも同一ディレクトリに置いてください）。kagipdf.rb は実行可能（`chmod a+x`）にしておいてください。

つぎに style ディレクトリにある LaTeX スタイルファイル kagipdf_hooktail.sty, kagipdf_report.sty, kagipdf_thesis.sty, mydefinition.sty を環境変数 TEXIPUTS の通ったディレクトリに置いてください。または、style ディレクトリに環境変数 TEXIPUTS を通してください。そのためには、bash, ksh の場合、`~/.bash_profile` に

^{*1} <http://docutils.sourceforge.net/docs/ref/rst/restructuredtext.html>

^{*2} <http://hooktail.org/wiki/>

^{*3} Vine Linux 3.x の場合、インストールが必要なのは Docutils と lv のみです。lv は apt-get で入手可能です。

```
TEXINPUTS=$TEXINPUTS:~/kagipdf/style
export TEXINPUTS
```

などと記述します。

2.2 Cygwin

上記手順と同じです。実行可能属性はつけなくていいかもしれません。Cygwin 環境から W32TeX を使用する場合、style ディレクトリにある LaTeX スタイルファイルを、

```
C:\tex\share\texmf\ptex\platex\misc
```

などの LaTeX コンパイル時に読み込まれるディレクトリに置いてください。

3 使用方法

PDF 化したいテキストファイルのあるディレクトリに移動して

```
$ kagipdf.rb infile
```

のように実行します。ここで `infile` は reStructuredText 方式 (+ 物理のかぎプロジェクト拡張) で書いたテキストファイルです。画像ファイルを含む場合は、それらも同じディレクトリになければいけません。うまく行けば、コマンド一発で PDF ファイルが生成されます。このときの PDF ファイル名は、入力ファイルの拡張子を `pdf` に置き換えたものです。同時に、TeX ソースファイル (拡張子 `tex`) と DVI ファイル (拡張子 `dvi`) も出力します。

また、リダイレクト:

```
kagipdf.rb < stream
```

や、パイプ:

```
stream | kagipdf.rb
```

から起動することもできます。この場合、標準入力を入力ファイルとして処理し、出力される pdf ファイルのファイル名は `kagipdf_default.pdf` となります。

Cygwin で試用する場合、後のセクション「Cygwin で試用する場合のヒント」もお読みください。

4 オプション

`-o <string>` アウトプット pdf ファイル名を指定。

```
例) $ kagipdf.rb -o index.pdf infile
```

`-k <type>` 出力 LaTeX ファイルの文字コードをにセット。デフォルト値は Cygwin 環境で Shift_JIS, それ以外で EUC-JP。<type> には s, u, e のいずれかを指定。それぞれ文字コードを Shift_JIS, UTF-8, EUC-JP にセット。

```
例) $ kagipdf.rb -ks infile
```

`-q` メッセージを極力出力しない。

-c pdf ファイルのみ残して, tex ソースや dvi ファイルは削除。
-l 中間ファイルをすべて残す。
--notex platex コンパイルを実行しない。
例) \$ kagipdf.rb --notex infile
--nopdf dvipdfmx による pdf ファイル生成を行わない。
例) \$ kagipdf.rb --nopdf infile
--noeps EPS ファイルの生成を行わない。
例) \$ kagipdf.rb --noeps infile
--style <type> LaTeX のスタイルファイルを指定<type> には hooktail, report, thesis を指定可能。
デフォルトは hooktail。
例) \$ kagipdf.rb --style thesis infile
--help オプション一覧を示すヘルプを表示。
--version バージョン情報を表示。

なお, リダイレクトおよびパイプから kagipdf.rb を起動した場合, -q オプションを付けて実行したのと同じこととなります。

5 Cygwin で試用する場合のヒント

Cygwin に日本語 platex をインストールしておらず, W32TeX でコンパイルしたいときは,

1. TeX ファイル出力で kagipdf.rb の実行を止める
2. Cygwin から W32TeX を呼び出せるようにする

の二つの解決策があります。これらについて少し説明します。

5.1 TeX ファイル出力で kagipdf.rb の実行を止める

TeX ファイル出力までで止める場合, 実行時に --notex オプションを付け:

```
$ kagipdf.rb -notex infile
```

とすると良いでしょう。TeX ソースファイルができあがるので, お使いの W32TeX 環境で改めてコンパイルしてください。Cygwin 上で kagipdf.rb を実行した場合, 出力 TeX ソースファイルの文字コードは自動的に Shift_JIS にセットされるので, そのまま W32TeX でコンパイルすることができます。他の文字コードで TeX ソースファイルを出力したい場合は, -k オプションで指定してください。

5.2 Cygwin から W32TeX を呼び出せるようにする

こちらの方が便利な解決策です。Cygwin から W32TeX を呼ぶには,

```
C:\cygwin\etc\profile.d\lilypond-profile.sh
```

というファイルの 74 行目あたりにある

```
if [ -z 'echo $TEXMF | grep "$datadir"' ]; then
```

```
TEXMF="{ $datadir, "kpsexpand \\$TEXMF'" }"  
export TEXMF  
fi
```

をコメントアウト（行頭に # を追加）してください。そして:

```
$ kagipdf.rb infile
```

のように実行してください。

6 変換処理の手順

物理のかぎプロジェクトで使用している「拡張^{*4}を加えた reStructuredText ファイル」:

```
foo.txt
```

から, 本来の仕様に従ったファイル:

```
foo.reST
```

を作ります。それを `rst2latex.py` でコンパイルして TeX ソースを出力, さらにプリアンプルなどに修正を加えて:

```
foo.tex
```

を生成します。その後, `platex` でコンパイル, `dvipdfmx` で PDF ファイルを生成します。

また, `figure` および `image` ディレクティブにより画像を挿入している場合は, `png` および `gif` については `eps` ファイルに変換して, `jpg` についてはそのまま PDF ファイルに取り込みます。

7 既知のバグ

- 拡張命令のエスケープがうまくいっていない部分がある
- `replace` ディレクティブを経由して図を挿入した場合, EPS に変換されない
- reStructuredText 的コメントアウトが一部通用しない (例: `image` ディレクティブ)

8 ToDo

- `raw` ディレクティブ中では拡張機能を OFF にする
- 画像ファイルが別ディレクトリでも大丈夫なようにする
- タグそのものを出力できるようにする
- 出力レイアウトの選択子 (レポート, 論文, プレゼン等) を洗練する
- 「@@ reference: @@」に対応させる
- 「@@ information: @@」に対応させる

^{*4} TeX 数式命令呼び出しの簡素化等。詳細は <http://tinyurl.com/czukc> を参照してください。

9 更新記録

- [2006-01-26 Ver.0.4.0] `--style thesis` の挙動をより論文向けに修正。 `\ ref{}` 命令を直接書けるようにした。 `kagipdf_thesis.sty` のレイアウトを修正。 LaTeX 命令定義用クラス `mydefinition.sty` を常に読み込むようにした。 `mydefinition.sty` に定義を二つを追加。 EPS ファイルへの変換を行わないオプション `-noeps` を追加。 README への加筆修正。
- [2005-12-25 Ver.0.3.4] JPEG から EPS ファイルへの変換ミスを修正。
- [2005-10-24 Ver.0.3.3] 入力ファイルの文字コードが UTF-8 だと LaTeX コンパイルに失敗するのを修正。 URL エンコードを含むハイパーリンクが上手く働かないのを修正。 URL エンコードを含むハイパーリンクが上手く働かないのを修正。 Vine の rpm にある `dvipdfmx` は `jpg` の直接取り込みに問題があるため、JPEG も EPS に変換することに変更 (オプションを適切に指定すれば、それほどファイルサイズは変わらない)。 `sourceforge.jp` に「kagiPDF」としてプロジェクト登録。
- [2005-10-21 Ver.0.3.2] `-q`, `-c`, `-l` オプションを追加。リダイレクトおよびパイプからの起動時は `-qc` オプション指定と同様に動作するようにした。ソース中に \$ そのものを記述できるよう処理を追加。 `report` スタイルを追加。 `thesis` スタイルを追加。 `< align > ... </align >` タグを埋め込めるようにした。かぎしっぽ拡張の処理をしない `--strict` オプションを追加。
- [2005-10-13 Ver.0.3.0] シェルスクリプトと Ruby が混在していたものを、Ruby で統一した。処理方法を見直し、メンテナンス性、拡張性を高めた。 `platex` 実行に `-interaction=nonstopmode` オプションをつけ、エラーがあっても途中で止まらないようにした。TeX プリアンプルに埋め込んでいた命令を、別スタイルファイルとして用意。 `< tex >` タグの直後に改行がないと数式が上手く表示されないバグを修正。
- [2005-07-25 Ver.0.2.2] JPEG ファイルを `convert` コマンドで `eps` に変換するとサイズが大きくなるので、JPEG は変換せず直接 PDF に取り込む方針に変更。Adobe Reader で開いたときの初期状態を「幅に合わせる」に設定。マニュアルやコメント等の更新。
- [2005-07-24 Ver.0.2.1] 出力ファイル名を指定するオプション `-o` を追加。
- [2005-07-19 Ver.0.2.0] 脚注参照の番号がおかしくなるバグを修正。しおり、ハイパーリンクを有効にした。実行時に `platex` コンパイルしないようにするオプション `-n` を追加。出力 `tex` ファイル文字コード選択オプション `-k` を追加。
- [2005-07-16 Ver.0.1.0] 公開。