

L^AT_EX 処理自動化ツール Clut_TE_X について

@mod_poppo

2019 年 12 月 10 日

1 Clut_TE_X とは

1.1 主な特徴

Clut_TE_X は、私 (@mod_poppo) が作っている、比較的新しい L^AT_EX 処理自動化ツールである。主な特徴は、

- ユーザーのディレクトリに .aux や .log のような「余計な」ファイルを作らない (最大の特徴)
- 相互参照等の解決のために自動で複数回処理を行う*¹
- p_TE_X 系列では自動的に dvipdfmx を実行して PDF ファイルを生成し、pdf_TE_X と同様の使い勝手を実現する*²
- 設定次第で Bib_TE_X や MakeIndex の実行にも対応する

である。

他の L^AT_EX 処理自動化ツールとの比較は、以下ようになる：

latexmk 複数回処理する、Bib_TE_X 等を自動で実行するという点では類似している。最大の違いは、Clut_TE_X が「余計なファイルを作らない (隠蔽する)」という点である。latexmk の `-auxdir` は MiK_TE_X でしか使えないほか、`-outdir` での §2.1 の問題を解決できていない。

ptex2pdf p_TE_X 系列でもコマンド一発で (dvipdfmx を実行して) PDF を生成するという点で、Clut_TE_X は ptex2pdf の機能面での上位互換となる。

arara, llmk これらのツールは処理手順を明示するスタイルなのに対し、Clut_TE_X は相互参照の解決や Bib_TE_X や MakeIndex の実行を必要に応じて自動的に行う。

(ツールではないが) MiK_TE_X Clut_TE_X は MiK_TE_X の `--aux-directory` オプションを T_EX Live で再現する試みと言える。

*¹ 処理を一回だけ行うように設定することも可能である。

*² もちろん、DVI 生成で止めておくことも可能である。

1.2 ユーザーのディレクトリに余計なファイルを作らない

L^AT_EX で文書进行处理すると、`.aux`、`.log` やその他 `.toc` や `.out` などのたくさんの補助ファイルが生成される。もちろんこれらのファイルは L^AT_EX にとっては意味のあるものだし、上級者ならこれらのファイルをトラブル時の原因究明に利用できるが、L^AT_EX のライトユーザーにとっては単なる余計なファイルに思えるだろう。

また、Dropbox 等のクラウドサービスの同期対象となるディレクトリで L^AT_EX 文書进行处理する際、補助ファイルが同期されてもあまり嬉しくはない。むしろ、同期されるファイルが増えて余計な通信が発生する、Web ブラウザーやモバイル端末から見た際にどれが重要なファイルかわかりにくい、などの弊害が考えられる。

他の自動化ツールには補助ファイルを削除する機能を備えているものがあるが^{*3}、各種パッケージが生成する補助ファイルを網羅しているとは限らない。それに、補助ファイルを削除してしまうと、その後の処理で相互参照等の解決がイチからやり直しになる。これに対して、Clut_TE_X は補助ファイルを「作らない」のではなく「ユーザーから見えない場所に作成する」ので、通常の実行では、相互参照等の解決のための処理回数が増えるということはない。

1.3 その他の特徴

■他のツールとの連携 L^AT_EX 処理自動化ツールに求められる機能は多数ある。そういう機能を片っ端から実装していると開発工数がいくらあっても足りない。よって、Clut_TE_X は「ユーザーのディレクトリに余計なファイルを作らない」ことをうまくこなすことを目指し、外部のツールとの連携で実現できることはなるべく外部のツールに任せる方針である。

例えば、「個々の文書用の設定（例：エンジンの種類、Br_TE_X や MakeIndex の使用の有無）を設定ファイルやマジックコメントに書けるようにする」という機能は、外部のツール（シェルスクリプト、Makefile、l_mmk、その他^{*4}）で実現できるので、わざわざ Clut_TE_X で実装する必要性は薄いと考える。

ただし、外部ツールからは取得の難しい情報に関しては、Clut_TE_X 側から提供できるようにする。例えば、Makefile を書くためには T_EX 文書が依存するファイルの一覧を取得できると便利だが、デフォルトでは Clut_TE_X は補助ファイルを隠蔽してしまうので、Clut_TE_X 外から依存するファイルの一覧を得るのは難しい。そのため、Clut_TE_X は `--make-depends` オプションによって依存先一覧を書き出せるようにしている。このほか、Clut_TE_X v0.4 では補助ファイルの出力先を得るための `--print-output-directory` オプションを実装した。

■使い勝手 Clut_TE_X では、T_EX 処理時のいくつかの設定のデフォルト値を変更している（`-interaction=nonstopmode` や `-file-line-error` などが有効）。このほか、エンジンが Lua_TE_X の場合は、標準出力に書き出されるメッセージを削減^{*5}するようにしている。

■処理時間の短縮のために 大規模な L^AT_EX 文書を書く場合は処理時間も馬鹿にならない。処理時間を短縮するための L^AT_EX 標準の機能としては `\includeonly` があるが、`\includeonly` を L^AT_EX 文書の冒頭に書いて

^{*3} latexmk の `-c` オプションなど。

^{*4} Clut_TE_X 専用のこういうツールとしては、@Ishotihadus 氏が houcluttex <https://github.com/Ishotihadus/houcluttex> というプログラムを公開している。

^{*5} 具体的には、`\input` されるファイル名の出力を抑制している。現時点では、この挙動を変更するオプションは提供していない。

て処理し直すというのは面倒である。そこで、Clut \TeX は `\includeonly` をコマンドラインオプションとして指定する機能を提供している。

2 動作の仕組み

2.1 `-output-directory` を使う上での困難

Clut \TeX の仕組みはざっくり言うと

1. テンポラリディレクトリ以下に適当にディレクトリを用意する
2. `-output-directory` を指定して \LaTeX を実行する
3. ($\text{p}\TeX$ 系列の場合は) `dvipdfmx` を実行して PDF ファイルを生成する
4. 生成された PDF ファイルまたは DVI ファイルをユーザーの手元にコピーする

となる。だが、 \LaTeX の `-output-directory` への対応状況はあまり良いとは言えない。

- `\include` でサブディレクトリのファイルを読み込んだ場合、対応する `.aux` ファイルの書き出しに失敗する。(MiK \TeX の `--aux-directory` ではこういう場合に自動でディレクトリを掘ってくれるようだ)
- 外部コマンド実行を伴うパッケージや Lua を使うパッケージでは、一般論としてうまくいかない。 \LaTeX パッケージから `-output-directory` の値を取得する手段があればパッケージ側で対応することもできたかもしれないが、筆者の知る限りそういう手段はない。

`\include` に関しては、「出力先のディレクトリがないというエラーが出たらそのディレクトリを掘ってから再処理する」という方針で対処している。`--aux-directory` のような \TeX エンジンに手を加える方式と比べると、処理回数が増えるのが難点である。

外部コマンド実行を行うパッケージに関しては、基本的に個別対応となる。`minted` や `epstopdf` のような一部のパッケージはパッケージオプションとして `-output-directory` の値を与えられるようになっているので、Clut \TeX からその値を渡せば良い。それ以外のパッケージに関しては、できることは多くない。

2.2 相互参照等のための複数回処理

省略。

3 今後の計画

Clut \TeX はまだまだ未熟なソフトウェアであり、実装したい機能はたくさんある。私が検討している機能の例を以下にいくつか挙げておく。

設定ファイルによるカスタマイズ ターミナルの色付けやビューワーの設定などの、文書の内容に影響しない項目をカスタマイズする手段を提供したい。また、現状では `--output-directory` を省略した場合に補助ファイル等はテンポラリディレクトリ以下に作られるようになっているが、他の場所（ホームディレクトリ等）を指定できるようにしたい。

より多くのパッケージへの対応 外部コマンドを実行するパッケージに関しては、現状では `minted` や `epstopdf` 等の `-output-directory` を考慮したパッケージしか対応できない。Clut $\text{T}_{\text{E}}\text{X}$ 用の補助 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ パッケージを作成して、`-output-directory` を考慮していないパッケージへパッチを当てるのができないか考えたい。

処理の成功時、失敗時のアクション 監視モードで処理の成功時に自動でビューワーを立ち上げたり、失敗時に何らかの通知を行う機能があると良い。

もちろん、ユーザーの皆さんからの要望も受け付けている。GitHub に Issue を立てるなり、Twitter で伝えるなり、 $\text{T}_{\text{E}}\text{X}$ Conf の会場で私に直接伝えるなりして頂きたい。

4 関連リンク

- GitHub <https://github.com/minoki/cluttex>
- $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ 処理自動化ツール Clut $\text{T}_{\text{E}}\text{X}$ をリリースした <https://blog.miz-ar.info/2018/10/cluttex-release/> 2018年10月13日
- $\text{T}_{\text{E}}\text{X}$ 実行の自動化ツールを作った (Clut $\text{T}_{\text{E}}\text{X}$) <https://blog.miz-ar.info/2016/12/cluttex/> 2016年12月28日
- 同人誌「 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ 文書処理自動化ツール Clut $\text{T}_{\text{E}}\text{X}$ 使い方とその仕組み」<https://lab.miz-ar.info/cluttex-book/>