

計算機基礎レポート

実験者番号 38 鄭潤賢

1 概要

この文書は計算機基礎レポート用に作成したプログラムを説明するための文書である．当初はこのプログラムをレポートの解答にする予定だったが，`stex` 上で動作しなかったため，正式な解答として選ぶことは断念した．しかし，せっかく作ったので報告のみ行なう．

今回はこの文書の作成にも利用している \LaTeX の機能を補うプログラムを作成した．補う手段は様々ではあるが，今回は私の環境に配慮して，コンソールアプリケーションを開発した．ただし，一部 GUI を利用する処理も存在する．アプリケーションの主な機能は \LaTeX による文書作成用のプロジェクトの生成，テンプレートファイルの管理，画像の添付などの形式的なコードの自動生成である．

2 背景

このアプリケーションを作成した背景は，長い春休みを経て久しぶりに実験レポートを描く際に，その構成やファイルの整理方法に少々苦心したためである．より手軽に，より整頓されたディレクトリ構造とその構造を用いたソフトウェアの連携が出来ないかと思い，今回のプログラムを作成した．また， \LaTeX という素晴らしい文書作成ツールを友人に勧める際に彼らの第 1 の懸念材料としてあがっていた，「使い方が複雑」という問題を少しでも解消したいという思いも，このプログラムの作成にいたった動機の 1 つである．

3 機能

ソースコードはこの文書と一緒に添付した ZIP ファイル内にあり，使い方については README.md ファイルに記載されている．また，https://github.com/jjj999/latex_launcher/ に ZIP ファイルと同様の内容のリポジトリが公開されている．

プログラムを利用する際は，まず入手したリポジトリ下にある `unix` または `win` ディレクトリから自身の環境を選択し，選択したディレクトリ下にある `bin` ディレクトリを環境変数の `PATH` に設定する．パスが通った後に

```
$ texl setup
```

を実行し，設問に答えることでセットアップが完了する．セットアップ時にはホームディレクトリ下に `texl` ディレクトリが生成され，そのディレクトリには設定ファイルや後述するテンプレートファイルなどが格納され，それらのファイルを利用してプログラムを実行していく．

プログラムの主要な機能については以下で述べる．それ以外の機能については添付したファイルを参照のこと．

3.1 プロジェクトの生成

プログラム内では \LaTeX 文書作成時に所定のコマンドを実行することにより生成されるディレクトリのことをプロジェクトと呼んでいる．このプロジェクトをコマンドを実行することによって生成するのがこのプログラムのメインの機能である．プロジェクトを作成するとプロジェクトディレクトリ下には現時点では `src`, `data`, `fig` の 3 つのディレクトリが置かれ，`src` ディレクトリ下には `[プロジェクト名].tex` ファイルが置かれる．基本的にはこの初めに生成

される tex ファイルを編集し、文書を作成していく。

プロジェクト作成時に生成される tex ファイルは、.texl ディレクトリ下にあるテンプレートファイルの複製である。テンプレートファイルの設定方法については次節で述べる。セットアップ終了後の初期状態ではデフォルトテンプレートが用意されており、自動的にデフォルトテンプレートのコピーが生成される設定になっている。ちなみにデフォルトテンプレートではプロジェクトを作成すると、著者名にセットアップ時に設定した自分の名前が、画像ファイルのパスにはプロジェクト生成時に同じく生成された fig ディレクトリのパスが設定されるようになっている。

3.2 テンプレートファイルの管理

プロジェクト作成時にはテンプレートファイルの複製が生成されることを述べたが、テンプレートファイルは複数管理出来る。管理する際には自身が作成したテンプレートファイルとその別名 (report など) をセットで登録し管理する。別名がキーであり、ファイルのパスが値となる。ここでテンプレートファイルがユーザーによって移動される場合を考慮して、テンプレートファイル登録時にそのファイルを.texl ディレクトリ下にコピーする。よって管理対象となるのは.texl ディレクトリ内にあるテンプレートファイルである。

テンプレートファイルには「現在のテンプレート」という概念を便宜上導入する。「現在のテンプレート」とは優先的にテンプレートファイルとして指定されるテンプレートのことを指し、プロジェクト生成時に明示的に指定しない限り、現在のテンプレートが tex ファイルとして生成される。よって、ユーザーが所望のテンプレートでプロジェクトを生成する方法は 2 通りあり、1 つは現在のテンプレートにそのテンプレートを設定し生成する方法、もう 1 つはプロジェクト生成時に明示的にテンプレートの別名を指定して生成する方法である。

テンプレートは自由に登録したり消去したりすることは可能なので、ユーザーが初めに用途別にいくつかのテンプレートファイルを生成してわかりやすい名前前で登録しておけば、素早くプロジェクトを生成できる。ただし、デフォルトテンプレート (キーは default) は消去不可能である。また、テンプレートの管理情報は.texl ディレクトリ下の設定ファイル (setting.json) に記載される。

3.3 画像の添付

この機能はプロジェクト作成後に文書を作成しながらターミナル上で行なうことを想定した機能である。画像の添付は表の添付とは異なり、(私の使う範囲では) 形式的に行われる場合が多いため、画像の名前を引数として指定することで、コピー&ペーストで画像の添付を行える L^AT_EX コードを自動生成する機能を作成した。この機能は現時点では柔軟性にかける部分が多々あるが、2 枚以上の画像を指定すると自動的に subfigure としてコードを生成する機能を備えている。1 発のコマンドで理想のレイアウト設定を指定することは極めて難しいため、このように枠組みだけは提供し、細かい設定はユーザーがレビューを見ながら調整することを期待し、あまり凝った実装は行わなかった。画像の添付機能はオプションを指定しなければコンソール上に自動生成されたコードが出力されるのだが、別ウィンドウにコードを表示するオプションを備えているため、コンソールが狭いような環境であっても、コピーするには十分な大きさのポップアップウィンドウが生成されるため比較的楽にコードを利用できる。

画像の添付処理ではコピー&ペーストしたコードが直ちにコンパイル可能なものであるために、プロジェクトのディレクトリ構造を利用している。すなわち、fig という画像専用のディレクトリを用意し、そのディレクトリ下に画像を置くことを強制することで、プロジェクト生成時にその fig ディレクトリのパスを画像参照先のパスとして設定した tex ファイル内で、自動生成されたコードが実行可能であるようにしている。

4 stex 上で動作しなかった原因

上で述べたプログラムは全て Python で記述している。よって Python がインストールされていない stex 上では動かないので、実行可能ファイルを Python スクリプトから生成し、それを動かすのはどうかと考え、実際に stex 上で作成した実行可能ファイルを実行してみた。しかし、「GLIBC_2.19」がないという表示とともにエラーが発生した。すなわち stex の glibc のバージョンが今回開発した私の PC のバージョンと異なっていたことが原因であると考えられる。