

Pandoc

发现了一个比较强的开源软件pandoc: <https://pandoc.org/>, 可以实现markdown, tex, html等文件的互转。

```
$ pandoc README.md -f markdown -s -o test.pdf --pdf-engine xelatex.exe # 生成pdf
$ pandoc README_zh_CN.md -f markdown -s -o test.tex # 生成tex
```

我实际测试了一下, 将.md文件转换为.pdf文件, 效果如下:

1. 英文版markdown

```
$ pandoc README.md -f markdown -s -o test.pdf --pdf-engine xelatex
```

DSPFFT

Description

The first curriculum design of Digital Signal Processing, FFT

Supported Compiler

Compiler	Platform	Target	C++11	C++14	C++17
GCC 7 ~	Linux	-m64	Perfectly	Perfectly	Perfectly
GCC 11	x86-64		supported	supported	supported
Clang 12	Linux	-m64	Perfectly	Perfectly	Perfectly
	x86-64		supported	supported	supported
MSVC	Windows	x64	Not	Supported	Supported
19.2	(64-bit)		supported		
MSVC	Windows	x86	Not	Compiler	Supported
19.2	(64-bit)		supported	warnings	

How to build

Linux

- Build library

```
make build [options]
```

The options can be:

- COMPILER: Specify the compiler you want to build with. The value can be `g++` or `clang++`. `g++` by default.
- CPP_STANDARD: Specify the C++ language standard. The value can be `-std=c++11`, `-std=c++14`, `-std=c++17`, `-std=gnu++11`, `-std=gnu++14`, `-std=gnu++17`, etc. The default value is `-std=c++11`.
- OPTIMIZATION: Specify the level of optimization. The value can be `-O0`, `-O1`, `-O2`, etc. `-O2` by default.
- WARNING_LEVEL: Specify the warning level. The default value is `-Wall -Wpedantic -Wextra`
- PREDEFINED_MACRO: Specify predefined macros, such as `-DUNICODE`. Empty by default.

Example:

```
make build CPP_STANDARD=-std=c++17 COMPILER=clang++
```

Then the library will be in the `build` directory. The binary library will be `build/bin/libdspfft.a` and its header file will be in the `build/include` directory.

2. 中文版markdown

```
pandoc.exe README_zh_CN.md -f markdown -s -o test_zh.pdf --pdf-engine
xelatex -V mainfont="SimSun"
```

DSPFFT

概述

数字信号处理第一次课程设计——快速傅里叶变换

支持的编译工具

编译工具	平台	目标平台	C++11	C++14	C++17
GCC 7 ~	Linux	-m64	Perfectly supported	Perfectly supported	Perfectly supported
GCC 11	x86-64				
Clang 12	Linux	-m64	Perfectly supported	Perfectly supported	Perfectly supported
	x86-64				
MSVC 19.2	Windows (64-bit)	x64	Not supported	Supported	Supported
MSVC 19.2	Windows (64-bit)	x86	Not supported	Compiler warnings	Supported

如何构建

Linux

- 构建静态库

```
make build [options]
```

options 可以是:

- COMPILER: 指定构建所用的编译工具。值可以是 g++ 或 clang++, 默认为g++
- CPP_STANDARD: 指定 C++ 语言标准, 值可以是 -std=c++11、-std=c++14、-std=c++17、-std=gn 等。默认为 -std=c++11
- OPTIMIZATION: 指定优化等级。值可以是 -O0、-O1、-Os 等, 默认为 -O2
- WARNING_LEVEL: 指定警告等级。默认值为 -Wall -Wpedantic -Wextra
- PREDEFINED_MACRO: 指定预定义宏, 例如 -DUNICODE, 默认为空

例如:

```
make build CPP_STANDARD=-std=c++17 COMPILER=clang++
```

然后, 生成的库将会位于 build 文件夹内。二进制库文件为 build/bin/libdspfft.a, 头文件位于 build/include 文件夹内。

(可以看到不是很美观hh)

所以我们需要先生成一个tex文件

```
$ pandoc README_zh_CN.md -f markdown -s -o test.tex
```

对这个tex文件需要进行一些**手动修改** (在开头加一句 `\documentclass[UTF8]{ctexart}`), 再使用以下工具编译:

```
$ latex test.tex
$ dvipdfmx test.dvi
```

随后生成一个稍微好看一点的pdf:

DSPFFT

概述

数字信号处理第一次课程设计——快速傅里叶变换

支持的编译工具

编译工具	平台	目标	C++11	C++14	C++17
		平台			
GCC 7 ~	Linux	-m64	Perfectly	Perfectly	Perfectly
GCC 11	x86-64		supported	supported	supported
Clang 12	Linux	-m64	Perfectly	Perfectly	Perfectly
	x86-64		supported	supported	supported
MSVC	Windows	x64	Not	Supported	Supported
19.2	(64-bit)		supported		
MSVC	Windows	x86	Not	Compiler	Supported
19.2	(64-bit)		supported	warnings	

如何构建

Linux

- 构建静态库

```
make build [options]
```

options 可以是:

我们可以在我们的 latexhelper 中加一个功能 `latexhelper pandoc xx.md --settings xx`, 将上述的这些命令行操作和文本编辑操作封装起来, 对pandoc导出的.tex文件进行自动编辑, 实现文档美化, 添加目录, 添加标题编号等功能。