The `c13n` System Backend Introduction

rne

Sep 20, 2024

Chapter 1

About the `md2tex` utility

1.1 Features

This is a Markdown to (La)TeX parser implemented in C, based on MD4C, with the following features:

1. **Compliance**: It is compliant to the latest version of CommonMark specification thanks to MD4C. Currently, we supports CommonMark 0.31.
2. **Extensions**: It supports some widely used extensions, namely: table, strikethrough, underline, and TeX style equations.
3. **Lenience**: It follows completely the GIGO philosophy (garbage in, garbage out). It sees any sequence of bytes as valid input. It will not throw any error when parsing.
4. **Performance**: It is very fast, parsing most of our posts in less than 1 ms.
5. **Portability**: It is tested to run on Mach-O, iOS, and Linux. It should run on Windows, BSDs, and all other POSIX-compliant OSes which has a working C compiler.
6. **Encoding**: It understands UTF-8 to determine word boundaries, and case-insensitive matching of a link reference label (Unicode case folding). However, it will *not* translate HTML entities and numeric character references.

1.2 Build Instructions

1.2.1 POSIX-compliant OSes

Make sure you have a working C compiler (Clang/GCC/...) and standard library (GLibC/musl/...), together with GNU Make and sed tools.

Then simply run make to generate the executable `md2tex`.

1.2.2 Windows

You can install Cygwin and follow the same step, or you can use the fragile way.

```
[language=sh]
cc -o md2tex md2tex.c md4c.c -O2 -Wall
```

1.3 MarkDown Spec

Generally, you should conform to the CommonMark spec when writing posts. However, because of the additional extensions, there are some extra rules.

1.3.1 Table

It supports GitHub-style tables .

Note that in the generated (La)TeX manuscript, the align information of the columns are currently ignored, and default to left align. It may be supported in the future.

1.3.2 Strikethrough

It supports `~delete~` (~~delete~~).

1.3.3 Underline

Underscore denotes an underline instead of an ordinary emphasis or strong emphasis. Thus, to get *italics* and **bold**, use `*` instead.

1.3.4 TeX-style Equation

TeX like inline math spans (`$...$`) and display math spans (`$$...$$`) ate supported. You are not required to escape ordinary dollars signs in most cases.

```
[language=Markdown]
This is an inline math span: $a + b = c$.

This is a display math span:
$$
  a + b = c.
$$

This is a dollar sign: $, 12$; equivalent to \$, 12\$.

This is a double dollar sign: $$; equivalent to \$\$.

Common knowledge: math spans cannot be nested.
$$foo $bar$ baz$$ is equivalent to \$\$foo $bar$ baz\$\$ which only bar is rendered as
```

```
Note: the opening delimiter or closing delimiter cannot be preceded or followed by an alphanumeri
x$a + b = c$ or $a + b = c$y will not be rendered as math (all $ rendered as \$)
```

Chapter 2

About the `drv.ltx` style sheet

It works only under LuaLaTeX.

2.1 Text style

We use `\underLine` and `\strikeThrough` to replace the LaTeX2e provided buggy `\underline` and the undefined `\del`.

2.2 Patch

Because SAX-like parser cannot elegantly capture the text between span delimiters, control sequence as listed below may appear in several circumstances.

```
[language=TeX]
\href{<url>}{} % from [](<url>)
\label{} % from an image that does not have a label
```

Instead of using further dark magic in the parser (i.e., modify the text callback), we handle this in LaTeX.

The control sequence `\href` is a hard one, because it relies on nasty catcode modifications to read in url containing chars that normally need to be escaped in a TeX manuscript, and thus means we cannot simply gobble the url into a parameter and further patch it. Also because of the complex infrastructure of hyperref, a custom `\href` has been implemented to solve this problem. It should have the exact functionality of `\href`, except when `#2` is empty, we supply one which has value `\url{#1}`. This command is LuaTeX specific.

```
[language=TeX]
\def\inner@ifempty#1{\begingroup\toks0={#1}\edef\p@r@m{\the\toks0}%
```

```
 \expandafter\endgroup\ifx\p@r@m\empty\expandafter\@firstoftwo\else%
 \expandafter\@secondoftwo\fi}
\def\inner@makeother#1{\catcode`#112\relax}
\def\href{\leavevmode\bgroup\let\do\inner@makeother\dospecials\inner@href}
\begingroup\catcode`[=1\catcode`]=2\catcode`\{=12\catcode`\}=12
 \gdef\inner@href{#1}{#2}[\pdfextension startlink user[/Subtype/Link/A<</Type/Action/S
```

Documenting this is unnecessary I think.

Without utilizing catcode dark magic, `\label` is a really easy one.

```
[language=TeX]
\begingroup\catcode`X=3\gdef\expnd@ifempty#1{%
 \ifX\detokenize{#1}X\expandafter\@firstoftwo\else%
 \expandafter\@secondoftwo\fi}\endgroup
\let\furui@label\label
\def\label#1{\expnd@ifempty{#1}\relax\furui@label{#1}}
```

2.3 Blocks

We include the `float` package as every float uses `[H]`. As graphics are represented using the `\image` control sequence in the parser, we have the following definition.

```
[language=TeX]
\setkeys{Gin}{width=.75\csname Gin@nat@width\endcsname,keepaspectratio}
\def\image#1{\includegraphics{#1}}
```

Another rather simple one is the thematic break `\thematic`.

```
[language=TeX]
\newcommand{\thematic}{\vspace{2.5ex}\par\noindent%
 \parbox{\textwidth}{\centering{*}\\[-4pt]{*}\enspace{*}\vspace{2ex}}\par}
```

2.4 CJK

Currently unimplemented, I think I shall first build the make system.

Chapter 3

The build system `make.py`

It's written in Python, and thus should be portable.

The only thing that should be mentioned about it is that please do not use `webp` as image format.