

Описание формата разметки mikiwiki

Базовые возможности

Михаил Уланов, mulanov@gmail.com

11 августа 2015 г.

Аннотация

Данный документ описывает основные возможности языка разметки *mikiwiki*. В документе описывается базовый синтаксис языка, и не рассматриваются следующие возможности: формулы, макросы, переменные, шаблоны, таблицы.

Содержание

1	Общие сведения	2
2	Основные конструкции mikiwiki	2
2.1	Комментарии	2
2.2	Абзацы и переносы строк	2
2.3	Заголовки и подзаголовки	3
2.4	Форматирование текста	3
2.5	Специальные символы и сочетания символов	4
2.6	Преформатированный текст	4
2.7	Ссылки	4
2.8	Списки	5
2.9	Нумерованные заголовки и абзацы	5
2.10	Иллюстрации	6
2.11	Экранирование	7
2.12	Препроцессор	7
3	Несовместимость с mediawiki	8

1 Общие сведения

Язык *mikiwiki* предназначен для разметки текстовых файлов для последующей конвертации их в различные форматы: HTML, $\text{\LaTeX}$, manpage и т. п.

В процессе разработки основными требованиями, предъявляемыми к формату были:

- удобство редактирования в текстовом редакторе;
- лёгкость конвертирования в различные форматы, в частности в HTML и в $\text{\LaTeX}$;
- по возможности совместимость с wikipedia и mediawiki.

В качестве отправной точки был выбран wiki-формат, используемый в проекте mediawiki, однако он был подвергнут некоторым улучшениям.

2 Основные конструкции mikiwiki

2.1 Комментарии

Строки файла, начинающиеся с символа “#” считаются комментарием и игнорируются. При этом после символа “#” не должно быть пробела и символов “#”, “*”, “:”, иначе строка будет интерпретирована как элемент нумерованного списка (для совместимости с mediawiki). Также комментариями не считаются директивы препроцессора `#warn`, `#define`, `#if`, `#else`, `#endif`, `#raw`, `#endraw`, `#include`.

При необходимости «закомментировать» большой многострочный фрагмент текста можно воспользоваться директивами препроцессора `#if 0` и `#endif`:

```
#if 0
текст,
который нужно закомментировать
#endif
```

2.2 Абзацы и переносы строк

Для разделения абзацев используется пустая строка, точно так же как в разметке mediawiki или $\text{\LaTeX}$.

Если необходимо начать новую строку, не прерывая абзац, используется символ “~” в начале строки, например:

В этом абзаце осуществляется перевод строки здесь: ~ продолжение абзаца на новой строке. Это - начало нового абзаца	В этом абзаце осуществляется перевод строки здесь: продолжение абзаца на новой строке. Это — начало нового абзаца
--	---

Строки входного файла можно «склеивать» в одну строку, размещая символ “\” в конце первой и последующих строк. Например, текст элемента списка должен располагаться на одной строке входного файла.

* первый элемент \ - списка не поместился \ - на одной строке * второй элемент	• первый элемент списка не поместился на одной строке • второй элемент
---	---

2.3 Заголовки и подзаголовки

Заголовки и подзаголовки выделяются символами “=”. Чем больше символов “=”, тем более вложен подзаголовок. Например:

```
= Заголовок книги =
== Заголовок главы ==
=== Подзаголовок ===
```

В отличие от mediawiki, закрывающие символы “=” ставить необязательно.

Заголовок должен располагаться на одной строке входного файла. При необходимости переноса строк во входном файле можно использовать символ “\” в конце строки:

```
== Длинный заголовок \
не поместился \
на одной строке
```

Заголовок первого уровня (определённый с одним символом “=”) не включается в оглавление, формируемое командой `${tableofcontents}`, но устанавливает переменную `${title}`, которая используется при формировании заголовка в формате HTML и титульного листа в формате L^AT_EX.

Возможно задание подзаголовка без формирования раздела:

```
~~ Подзаголовок ~~
```

Завершающие символы “~~” также необязательны.

Альтернативный способ задания подзаголовка:

```
=== Заголовок ===
===~ Подзаголовок ~===
```

Подзаголовок первого уровня (определённый последовательностью “=”) устанавливает переменную `${subtitle}`, которая используется при формировании заголовка в формате HTML и титульного листа в формате L^AT_EX.

2.4 Форматирование текста

Полужирный и курсивный текст кодируется различным количеством апострофов, так же как в mediawiki.

```
'''полужирный текст'''
''курсив''
''''полужирный курсив''''
```

Существует альтернативная возможность для обозначения полужирного, курсивного, подчёркнутого, перечёркнутого и моноширинного текста:

```
{*полужирный текст*}  
{/курсив/}  
{_подчёркивание_  
{-перечёркивание-}  
{|моноширинный текст|}
```

В этом случае «закрывающий» символ перед закрывающей фигурной скобкой не обязателен, например можно написать так:

```
{_подчёркивание}
```

Верхние и нижние индексы задаются комбинациями $\sim\{индекс\}$ и $_ \{индекс\}$ соответственно, например:

```
L_{Iso}=10^{\sim{50}} \text{ erg/s}
```

В отличие от TeX'а индекс нужно заключать в фигурные скобки, даже если он состоит из одного символа.

2.5 Специальные символы и сочетания символов

Неразрывный пробел обозначается символом “~”, аналогично TeX'у. При необходимости вставить в текст сам символ “~”, необходимо его заэкранировать: “\~”.

2.6 Преформатированный текст

Также как и в mediawiki, пробел в начале строки обозначает преформатированный моноширинный текст. При необходимости вставить пустую строку в преформатированный текст, используйте строку из символа “~”:

```
Это первая строка преформатированного текста.
```

```
Это вторая строка.
```

```
~
```

```
Это строка после пустой строки.
```

Внутри блока преформатированного текста не работают никакие команды форматирования, кроме подстановки значений переменных.

2.7 Ссылки

Гиперссылки задаются двойными квадратными скобками. При необходимости можно задать текст ссылки:

```
[[URL]]
```

```
[[URL Текст ссылки]]
```

Существует сокращённый формат ссылок на статьи Википедии. При этом необязательный текст ссылки отделяется вертикальной чертой.

```
[[w:Wikipedia article|текст ссылки]]
[[r:Статья в русской википедии]]
```

Можно задавать ссылки на внешние ресурсы в одинарных квадратных скобках (для совместимости с mediawiki), но при этом обязательно указывать протокол `http:`, `ftp:` или `mailto:`.

```
[http://www.wikipedia.org Википедия]
[ftp://ftp.sunet.se]
```

2.8 Списки

Поддерживаются нумерованные, ненумерованные списки, списки с отступом, списки определений, совместимые с mediawiki. Поддерживается неограниченная вложенность списков. Текст нумерованного списка должен быть отделён от значка “#” хотя бы одним пробелом, чтобы он не посчитался комментарием.

* Ненумерованный * Список	• Ненумерованный • Список
# Нумерованный # Список	1. Нумерованный 2. Список
* Вложенные списки *# Номер 1 *# Номер 2 * Второй вложенный список *# Номер 1 *# Номер 2	• Вложенные списки 1. Номер 1 2. Номер 2 • Второй вложенный список 1. Номер 1 2. Номер 2
;Термин:Определение : Просто строка с отступом :: С двойным отступом	Термин Определение Просто строка с отступом С двойным отступом

Текст элемента списка должен располагаться на одной строке входного файла. При необходимости переноса строк во входном файле можно использовать символ “\” в конце строки:

```
* длинный элемент \
списка не поместился \
на одной строке
```

2.9 Нумерованные заголовки и абзацы

Используя сочетание символа “%” и точек можно задать автоматическую нумерацию заголовков и подзаголовков. Каждый символ “%” будет заменён на соответствующее число. Если не ставить точку после символа “%”, то после номера раздела точки не будет (ГОСТ 2.105-95).

<pre>== %. Нумерованный заголовок === %..%. Подзаголовок == % Заголовок без точки после номера</pre>	1. Нумерованный заголовок 1.1. Подзаголовок 2 Заголовок без точки после номера
--	--

Подобным образом осуществляется и нумерация абзацев:

<pre>== % Заголовок % Нумерованный абзац === %..%. Подзаголовок % Первый нумерованный абзац % Второй нумерованный абзац %..%. Подпункт</pre>	1 Заголовок 1.1 Нумерованный абзац 1.2 Подзаголовок 1.2.1 Первый нумерованный абзац 1.2.2 Второй нумерованный абзац 1.2.2.1 Подпункт
--	--

При этом нумерация абзацев (пунктов) и заголовков сквозная, а к номеру пункта автоматически добавляется номер соответствующего заголовка.

Между символами “%” ставить точки необязательно. Важно только количество символов “%” и наличие или отсутствие точки в конце номера, чем определяется, будет ли поставлена точка после номера в выходном файле. Таким образом, следующие две записи эквивалентны:

```
== %..% Заголовок
== %% Заголовок
```

Нумерованные абзацы отличаются от нумерованных списков тем, что во-первых, нумерация абзацев привязана к нумерации заголовков, во-вторых, нумерованный абзац является полноценным абзацем: может состоять из нескольких строк и форматируется как абзац, в то время как списки обычно форматируются с дополнительным отступом слева.

При ссылке на нумерованный заголовок или абзац в текст ссылки автоматически вставляется номер заголовка или абзаца (без конечной точки):

<pre>== %. Заголовок \${label sec} %. Пункт. \${label point} Смотри п.\${ref point} в разделе \${ref sec}.</pre>	1. Заголовок 1.1. Пункт. Смотри п.1.1 в разделе 1.
--	---

2.10 Иллюстрации

Для вставки в текст изображения, имя графического файла задаётся в квадратных скобках. Если картинка не является фотографией из альбома, то перед именем файла картинки необходимо поставить один из символов “!”, “,”, “;”, “>”. После имени файла можно опционально указать ссылку, по которой будет осуществлять переход по клику на картинке (в режиме HTML).

[!image.png url/link.html] Просто картинка
[,image.png] Картинка снабжается подписью с номером рисунка
[;image.png] Картинка растягивается по ширине колонки текста
[>image.png] Картинка, выровненная по правому краю

2.11 Экранирование

Для того чтобы символы, выполняющие специальное действие, лишились этой особенности, их необходимо экранировать при помощи обратной косой черты “\”. Ниже рассматриваются символы, требующие экранирования.

- Обратная косая черта (\) требует экранирования всегда.
- Тильда (~) требует экранирования всегда.
- Фигурные скобки ({, }) требуют экранирования только в конструкциях выбора типа шрифта ({/, {*, {_, {!, {-) и верхних и нижних индексов (_{ }, ^{ }), причём в последнем случае достаточно заэкранировать либо только символ _ или ^, либо только фигурные скобки. В остальных случаях одиночные фигурные скобки экранирования не требуют.
- Знак доллара (\$) требует экранирования перед фигурной скобкой (\${).
- Одинарные квадратные скобки ([,]) используются либо для задания внутри абзаца ссылок, совместимых с mediawiki (в этом случае содержимое скобок должно начинаться с ftp:, http:, https:, mailto:), либо для задания картинок вне абзаца. В остальных случаях экранирование фигурных скобок не требуется.
- Двойные квадратные скобки ([[,]]) требуют экранирования всегда.
- Экранирование символов #, *, : требуется только при использовании их в начале строки, когда они могут быть расценены как определение элемента списка.
- Экранирование символа % требуется только при использовании его в начале строки, когда он может быть расценен как номер абзаца или заголовок.
- Одинарные апострофы (') экранирования не требуют, двойные и более требуют.
- Кавычки (") не требуют экранирования.

При экранировании символов, не имеющих специального значения, значение символа не меняется. Поэтому, если есть сомнение, нужно ли экранировать символ, экранируйте без опаски — хуже не будет. Однако следует помнить, что чрезмерное использование экранирующей обратной косой черты снижает читаемость исходного текста.

2.12 Препроцессор

2.12.1 #warn, #warning, #error

Директивы **#warn** и **#warning** являются синонимами. Текст, указанный в директиве выводится в поток STDERR при обработке исходного текста.

Директива **#error** выводит указанный текст в поток STDERR и аварийно завершает работу конвертера.

В выводимом тексте допускается использовать переменные, но не макросы. Например:

```
#warning Output format is ${format}.
```

2.12.2 `#define`

Директива `#define` позволяет определять переменные.

```
#define sun Солнце
#define var1 Пусть всегда светит ${sun}
#define var2
```

Подробнее переменные рассмотрены в полном руководстве по *mikiwiki*.

2.12.3 `#if`, `#else`, `#endif`

Директивы позволяют осуществлять условную конверсию исходного текста. Если значение аргумента директивы `#if` является истиной, то обработке подвергается текст до последующей директивы `#else` или `#endif`. Если значение ложно или неопределено, то обрабатывается только текст между директивами `#else` и `#endif`.

При помощи следующей конструкции можно организовывать многострочные комментарии.

```
#if 0
    строка
    строка
#endif
```

2.12.4 `#raw`, `#endraw`

Текст, заключенный между директивами `#raw` и `#endraw` включается в выходной файл без каких-либо преобразований. Не подставляются даже значения переменных.

2.12.5 `#include`

Файл, указанный в аргументе директивы `#include` включается в обработку в месте указания директивы.

3 Несовместимость с mediawiki

В mediawiki можно использовать произвольную HTML-разметку, например `<br>`, `<tt>`, ` `. Так как mwiki не является HTML-ориентированной, никакие HTML-тэги не поддерживаются.