
Содержание

1. Документирование программ	1
------------------------------------	---

1. Документирование программ

Сергей Юшинин <http://softarius.ru> <sergey@yushinin.ru>
07.02.2015

На определенном этапе развития программной системы неизбежно возникает задача разработки пользовательской документации. И тут возникает технический вопрос выбора форматов и инструментов разработки документации.

1.1. Выходные форматы

С выбором конечного формата обычно проблем не возникает, так как целевая операционная система предъявляет свои требования. Так, например для программ для Windows это формат скомпилированной справки CHM, для Linux и BSD систем это - man. Общим для всех систем форматом для онлайн справки является html, а для печати - pdf.

Ситуация усложняется в случае, если необходимо иметь документацию в нескольких форматах - для распространения с программой (chm или man), для размещения на сайте (html) и для печати (pdf). При этом возможно, что содержание документации в различных форматах может несколько отличаться. Например, видеофрагменты имеет смысл включать в онлайн документацию, а в печатной версии их нужно заменять на статическое изображение, возможно дополненным qrcode ссылкой на видеофрагмент.

Кроме того, содержание документов может отличаться и для различных категорий пользователей, версий, комплектов поставки и других факторов.

1.2. Исходные форматы

Не смотря на кажущуюся очевидность необходимости использования специально созданных программ и здесь не все так однозначно.

В зависимости от целевой операционной системы подходы отличаются.

1.2.1. Проприетарные исходные форматы

Так, для создания скомпилированной справки для Windows в формате chm фирма Microsoft предлагает использовать специальный бесплатный компилятор HTML Help Workshop. При этом исходные тексты должны быть подготовлены в формате html (редактор в поставку не входит), а файлы оглавления - в специфическом формате. Никаких средств формирования печатных руководств не предоставляется.

Разумеется, специализированные программы для создания справки (Robohelp, Help&Manual, HelpScribble и им подобные) предоставляют высокий уровень сервиса,

обладают возможностью формирования выходных документов в различных форматах и даже в некоторой степени профилировать содержимое.

Однако им присущи следующие недостатки:

1. Во-первых, все эти системы коммерческие и лицензируются по количеству используемых рабочих мест
2. Во-вторых, используемый ими внутренний формат является проприетарным и не поддерживается никаким ПО, кроме продаваемого. Возможность импортировать файлы в проект вам, конечно, будет предоставлена, а вот экспортировать проект в какой либо открытый формат, пригодный для дальнейшей обработки, не удастся. Даже в случае использования в качестве внутреннего формата XML (как, например, Help&Manual) схема его остается закрытой и никак не задокументированной.
3. В-третьих, возможности по изменению внешнего вида выходного документа являются недостаточными для формирования, например, документации в соответствии с требованиями ГОСТ.
4. В-четвертых, с этими программами организовать коллективную работу если и возможно, то крайне затруднительно

1.2.2. Простые форматы разметки

Рациональной альтернативой представляется применение простых и, как следствие, быстро осваиваемых форматов разметок.

На сегодняшний день таких форматов несколько:

- ASCIIDoc, используемый дефакто в Linux (BSD) системах;
- Wiki, применяемый в различного рода энциклопедиях и даже давший им общее название;
- Markdown - формат документирования систем в системе контроля версий Git.

Все эти разметки используют некоторый символьный нетеговый набор правил оформления заголовков, иллюстраций и ссылок, предполагающий редактирование в простых текстовых редакторах. Подготовка же пригодного к просмотру вида осуществляется программно как правило на стороне сервера.

Например, Википедия преобразует Wiki-формат в HTML "на лету". Веб-портал системы Git так же способен показывать документы в разметке Markdown в пригодном для чтения в браузере виде.

1.3. Редакторы

Не смотря на то, что для создания и редактирования исходных текстов достаточно возможностей блокнота, некоторые сервисные функции, такие как проверка правописания и подсветка разметки были бы писателю весьма кстати.

В статье <http://www.ixbt.com/soft/markdown-online-2.shtml> приведен обзор онлайн-редакторов поддержкой markdown-синтаксиса, а в <http://www.ixbt.com/soft/markdown-online-3.shtml> приведен обзор пяти настольных редакторов, поддерживающих формат markdown по умолчанию, так сказать "из коробки".

Одним из таких редакторов является MarkdownPad.

1.3.1. MarkdownPad

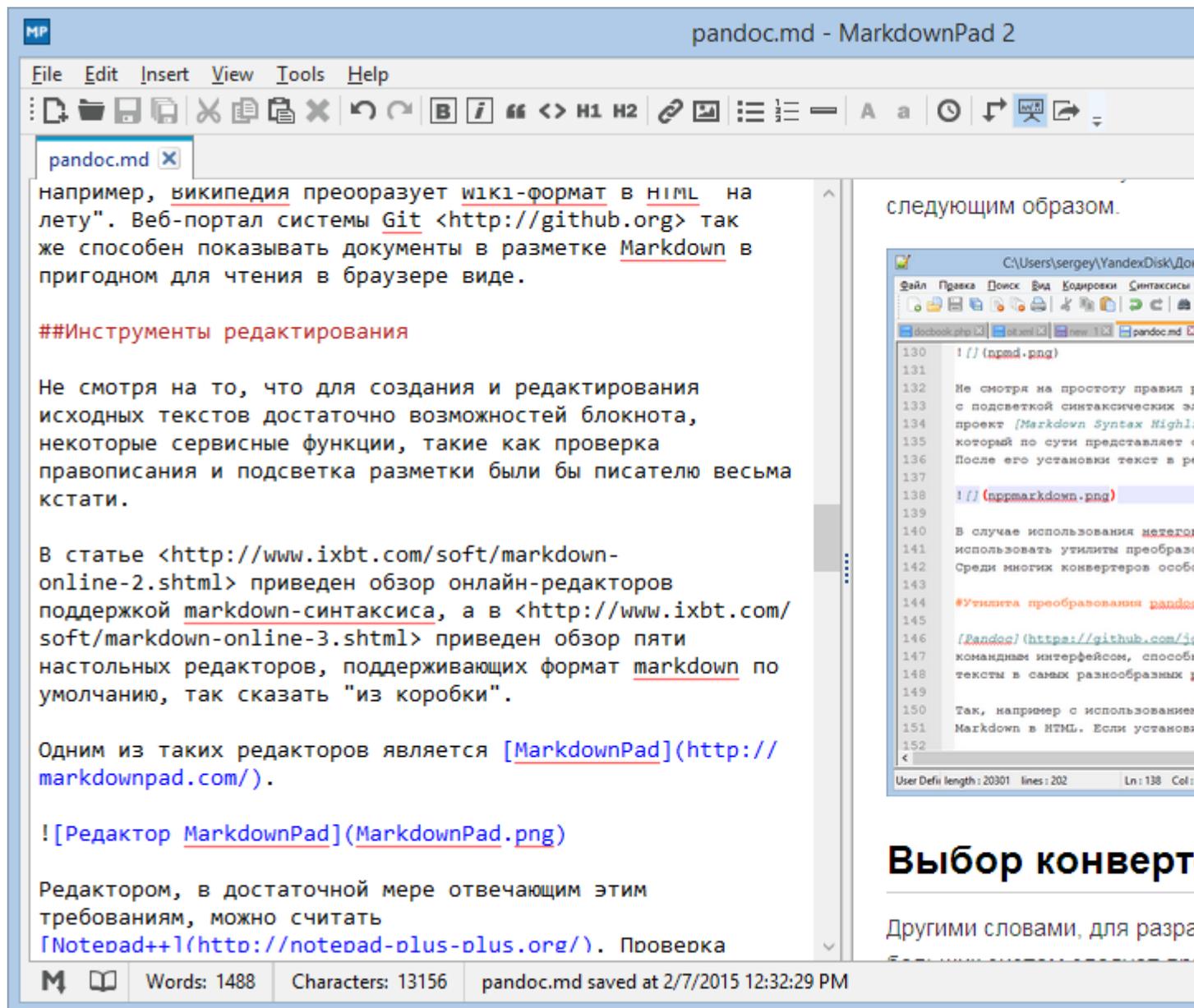


Рис. 1. Редактор MarkdownPad 2

Как видно из копии экрана редактор MarkdownPad 2 поддерживает "живой" предварительный просмотр редактируемого файла с поддержкой синхронной прокрутки исходного текста и результата рендеринга.

При установке на Windows 8 может возникнуть ситуация, когда предварительный просмотр недоступен.

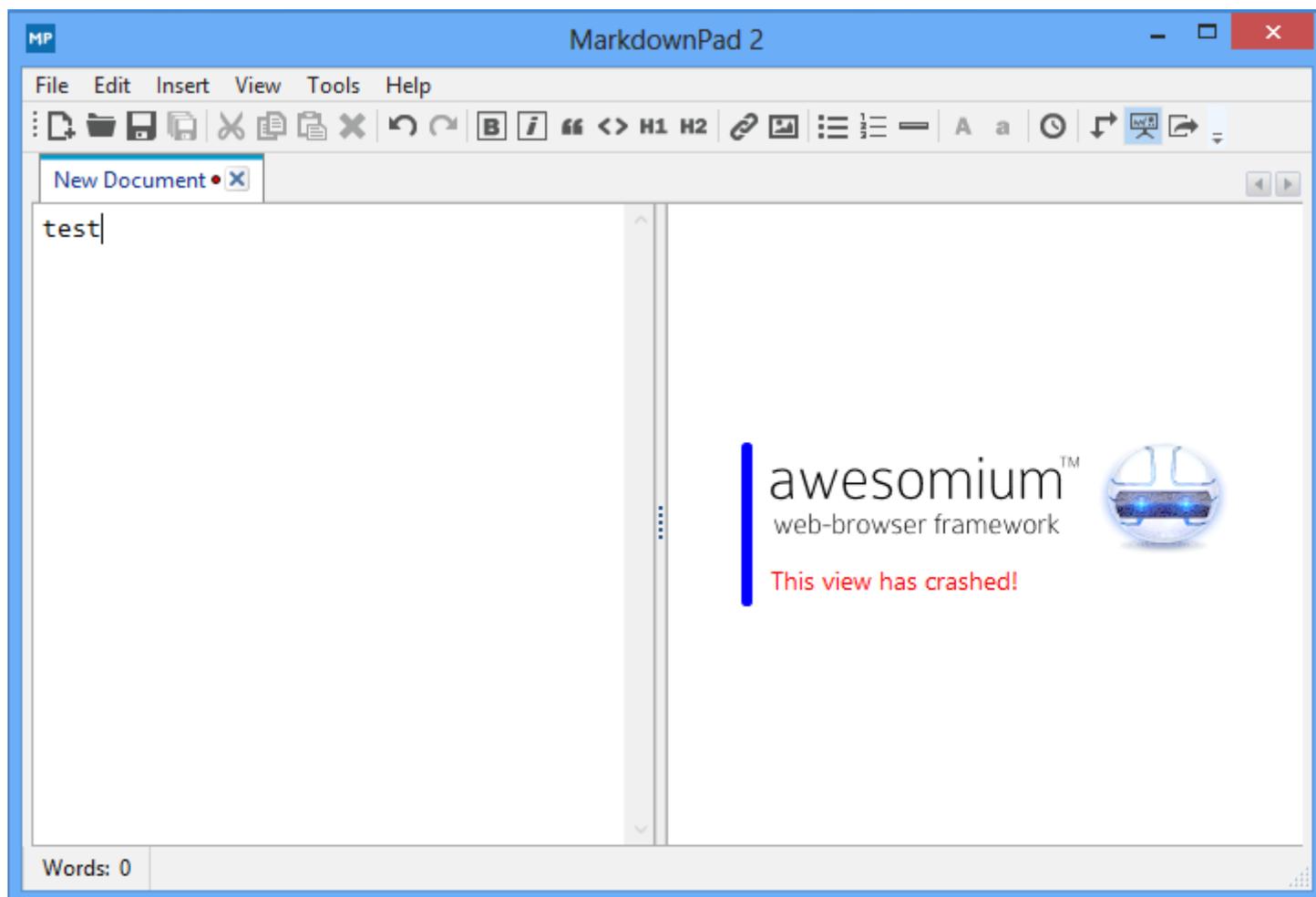


Рис. 2. Сообщение о крахе системы предварительного просмотра

По заявлению разработчиков <http://markdownpad.com/faq.html#livepreview-directx> это связано с необходимостью установки специфического SDK веб-рендеринга Awesomium 1.6.6 SDK, который в свою очередь использует DirectX.

Редактор поддерживает подсветку синтаксиса, проверку синтаксиса *одного* языка (в том числе русского), экспорт в форматы HTML, PDF (только в платной версии).

Иными словами, MarkdownPad 2, как и другие специальный редакторы, является хорошим выбором для технического писателя.

В тех же случаях, когда пользователю предстоит редактировать файлы различного формата, можно адаптировать свой редактор и для редактирования текстов с markdown-разметкой.

1.3.2. Notepad++

Редактором, в достаточной мере отвечающим этим требованиям, можно считать Notepad++. Проверка правописания многих языков поддерживается с помощью специ-

ального плагина. Причем поддерживается проверка текста *на нескольких языках одновременно*.

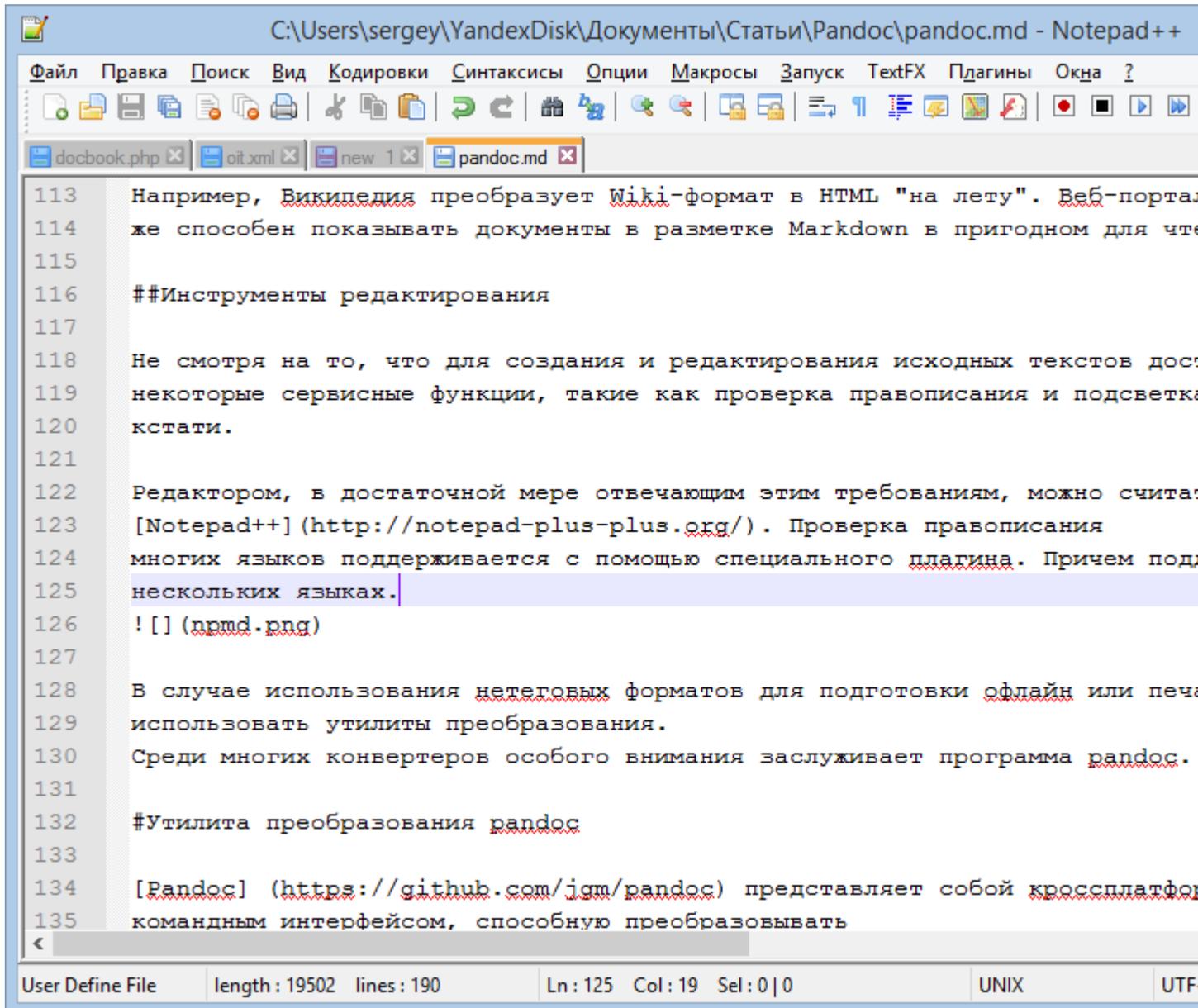


Рис. 3. Редактор Notepad++

Не смотря на простоту правил разметки автору текстов было бы удобней работать с подсветкой синтаксических элементов. Применительно к Notepad++ в этом поможет проект Markdown Syntax Highlighting for Notepad++, который по сути представляет собой конфигурационный файл пользовательского языка Markdown. После его установки текст в редакторе выглядит следующим образом.

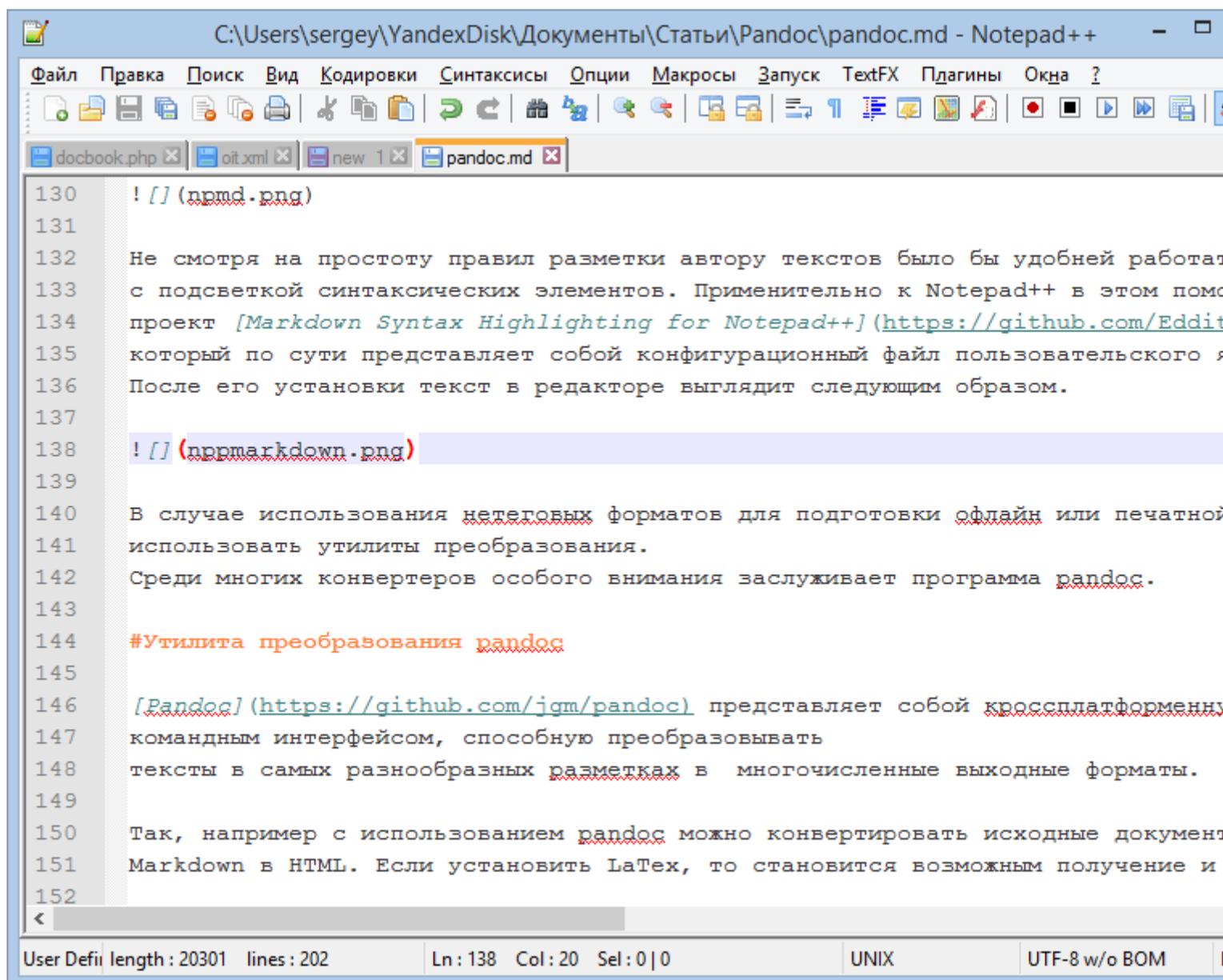


Рис. 4. Редактор Notepad++ с подсветкой элементов разметки markdown

1.4. Quota

Примечательно, что редакторы с поддержкой markdown существуют даже для мобильных платформ. На рисунке приведена копия экрана смартфона с запущенным редактором Quoda Code Editor.

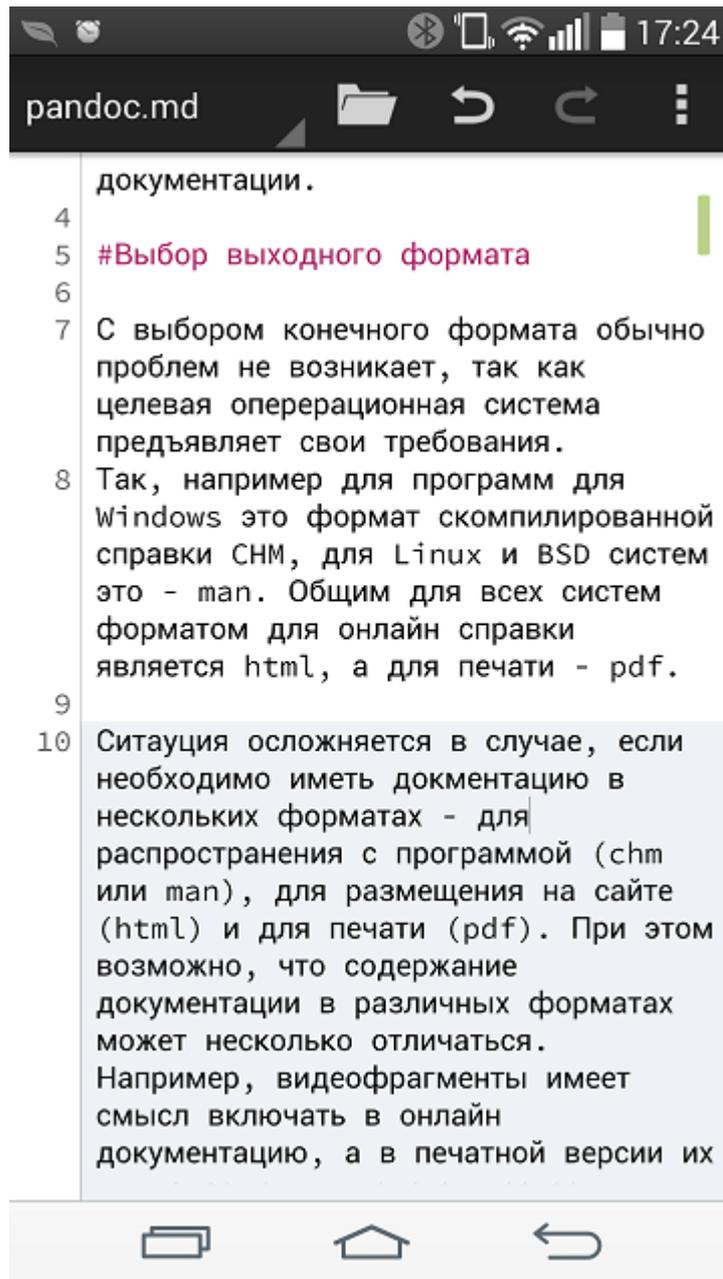


Рис. 5. Quoda Code Editor - универсальный редактор для Android с поддержкой разметки markdown

Следует сказать, что большая часть этой статьи набрана именно в этом редакторе, а уже потом выгружена на компьютер для доработки.

По результатам анализа возможностей языка разметки Markdown и специальных редакторов можно рекомендовать их применение для документирования систем средней сложности.

1.4.1. Открытые теговые форматы

Вместе с тем, для разработки программной документации больших систем следует применять в качестве исходного формата открытый, хорошо документированный формат. В качестве средства формирования - инструмент с широкими возможностями

по настройке внешнего вида, профилирования и способностью формировать документы в различных форматах.

Этим требованиям в полной мере отвечают такие системы как DITA и Docbook. Не смотря на некоторые различия, обе системы имеют много общего:

- используют в качестве исходного формата документированный (схематизированный) XML, что обеспечивает возможность использования для редактирования любого XML-редактора с функцией валидации;
- для конвертирования в один из результирующих форматов может быть использован практически любой xsl-преобразователь xslproc, xalan, saxon и др.;
- для получения pdf-документа используется промежуточный формат xsl-fo, из которого средствами любого fo-процессора (например, Apache FOP или XEP) уже формируется pdf;
- для настройки внешнего вида и профилирования используются многочисленные параметры преобразований, а в случае необходимости - добавлением пользовательских xsl-шаблонов.

Следует особо подчеркнуть, что данные системы используют семантическую разметку в исходных документах. Внешний вид же выходного документа определяется правилами и параметрами преобразований. Такой подход позволяет на этапе написания исходных текстов автору не задумываться над типографикой и дизайном, а сосредоточиться исключительно на смысловом содержании.

Вместе с тем практический опыт использования, в частности Docbook, подтвержденный и в ряде публикаций (<http://habrahabr.ru/post/212881/>), показал, что и при использовании столь продуманной технологии возникают некоторые сложности:

- создание исходных текстов в формате XML определенной схемы требует от технического писателя навыков работы со специальными редакторами;
- хорошие XML-редакторы с поддержкой Docbook - продукты коммерческие и не дешевые (например, oXygen XML Editor, Altova XMLSpy XML Editor);
- богатые возможности XML-разметки влекут за собой усложнение формата. Например, для вставки в текст иллюстрации с подписью в разметке Docbook необходимо использовать четыре вложенных тега.

Естественно, что вышеперечисленные недостатки сдерживают широкое применение XML-ориентированных технологий единого источника.

В случае использования нетеговых форматов для подготовки офлайн или печатной документации необходимо использовать утилиты преобразования. Среди многих конвертеров особого внимания заслуживает программа pandoc.

1.5. Утилита преобразования pandoc

Pandoc представляет собой кроссплатформенную программу с командным интерфейсом, способную преобразовывать тексты в самых разнообразных разметках в многочисленные выходные форматы.

Так, например с использованием pandoc можно конвертировать исходные документы в разметках ASCII doc, Wiki, Markdown в HTML. Если установить LaTeX, то становится возможным получение и PDF.

Так, например, преобразование исходного текста этой статьи в html формат можно выполнить следующей командой:

```
pandoc -f markdown pandoc.md -t html -o pandoc.html -H h.html
```

Результатом будет готовый html-файл:

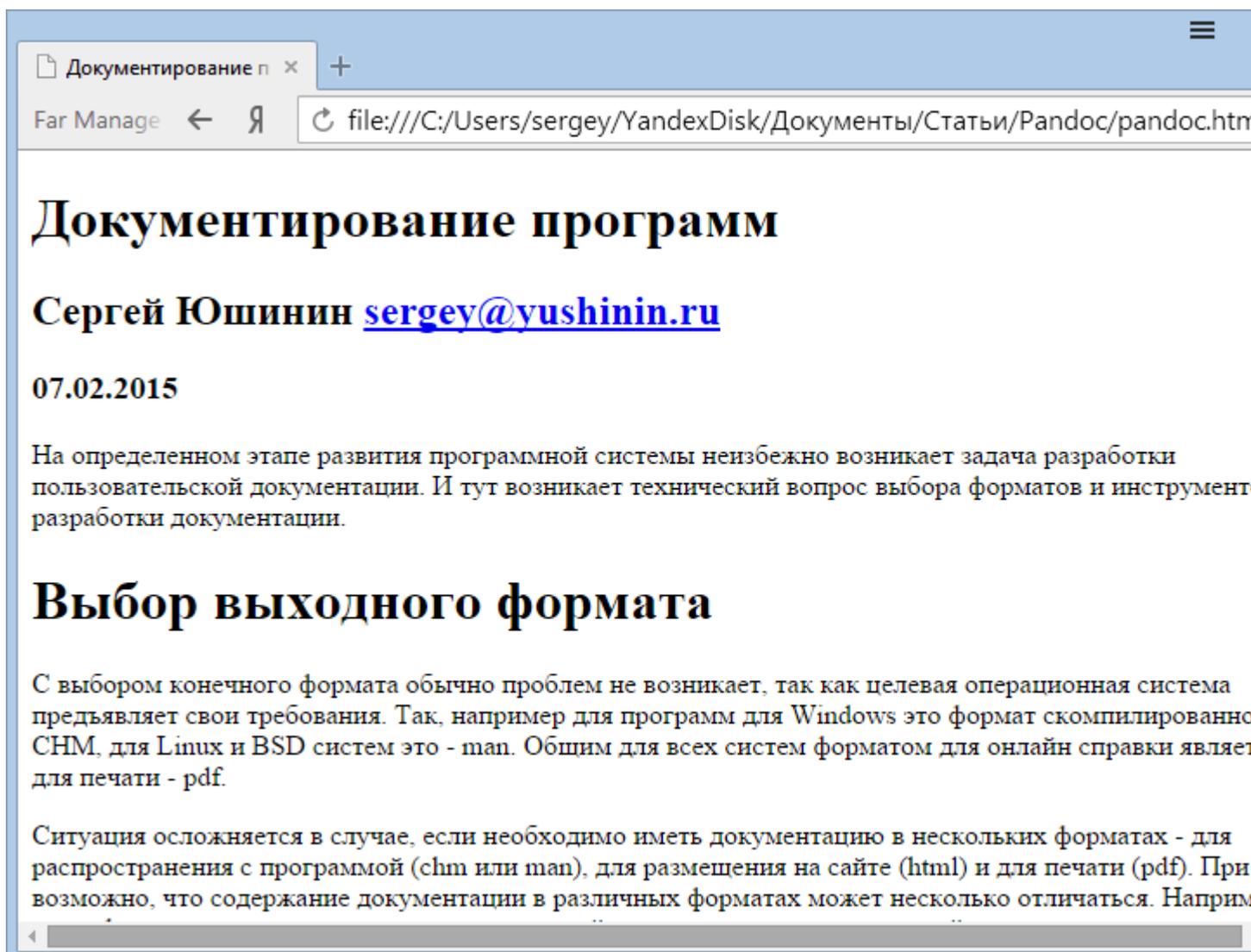


Рис. 6. HTML-документ, сформированный из Markdown утилитой pandoc

За свою универсальность программа образно названа автором "швейцарским армейским ножом".

Действительно, pandoc справляется с конвертированием без каких-либо потерь информации. При конвертировании из формата Markdown поддерживается чтение трех параметров метаданных - заголовка, автора и даты документа. Поддерживается так же передача параметров командной строки для установки некоторых специфических свойств, например языка документа. Есть возможность задать свой шаблон выходного документа, до некоторой степени видоизменяя его.

Так, например, в приведенном выше примере подразумевается, что в текущей папке есть файл h.html, который играет роль заголовка. Если в этом файле добавить ссылку на стилевой файл и определив `<base target="_blank">`, получим следующий результат:

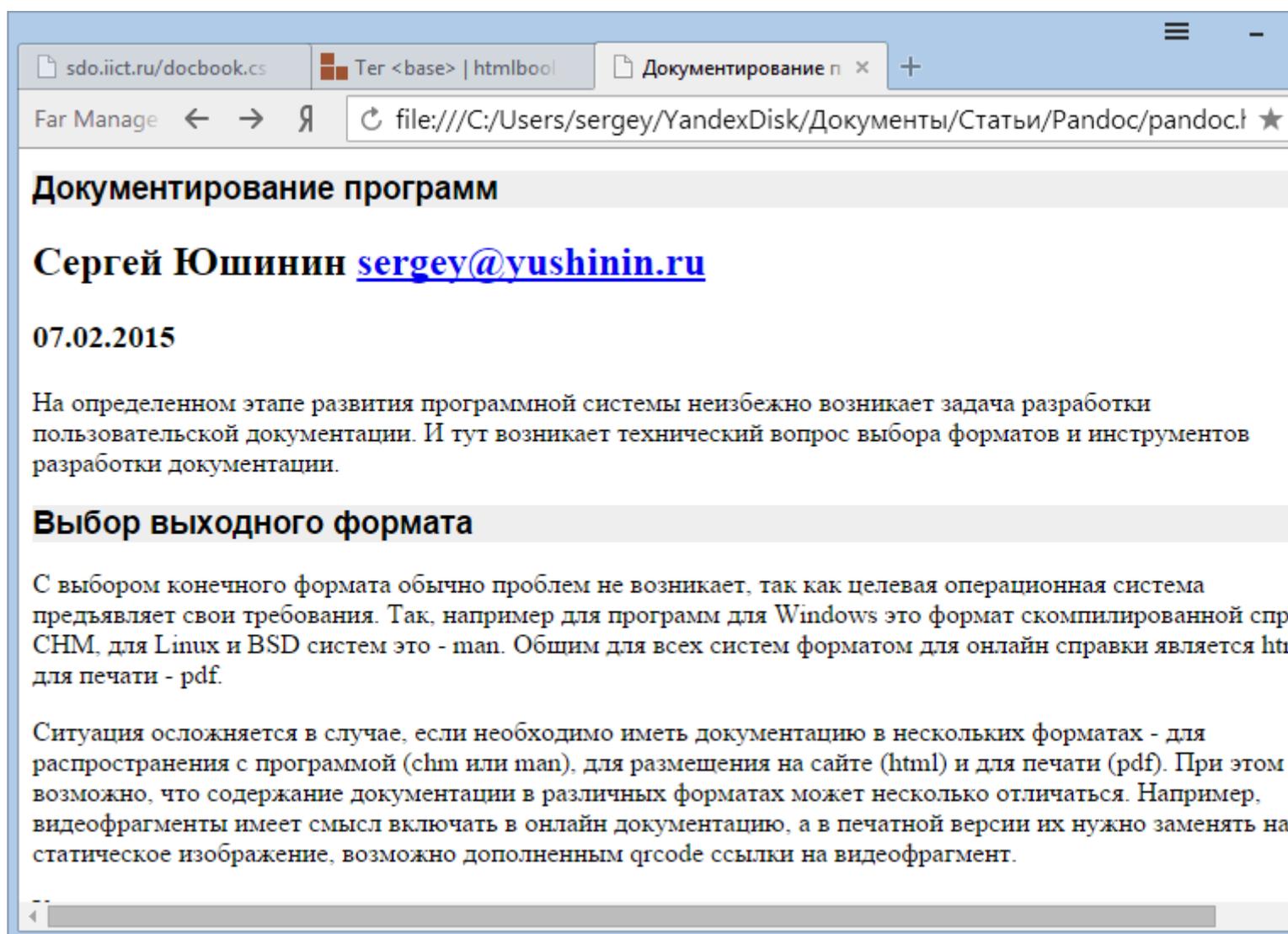


Рис. 7. HTML-документ, сформированный pandoc с использованием заголовочного файла со ссылками на стили

Как видно из примера, заголовки приобрели свой стиль, а внешние ссылки стали открываться в новой вкладке браузера.

Вышеописанные возможности формата делают оправданным использования разметки Markdown для документирования относительно небольших программных систем к оформлению которых не предъявляется требований ГОСТ, что и доказывается ее широким использованием в системе Git.

Что же касается больших систем с обширной и сложной документацией, то для ее создания видится применение системы единого источника Docbook. Могут иметь место и переходные случаи, когда масштаб проект проявляется не сразу.

1.6. Docbook

Сложность создания исходных XML-источников можно преодолеть путем использования исходных текстов в формате Markdown с последующим их конвертированием в Docbook. Такое преобразование поддерживается утилитой pandoc. Так, команда

```
pandoc -f markdown pandoc.md -t docbook -o pandoc4.xml -H h.xml
```

создаст результирующий файл.

Использование заголовочного файла `h.xml` (можно просто пустого) необходимо для корректной обработки метатегов и формирования статьи.

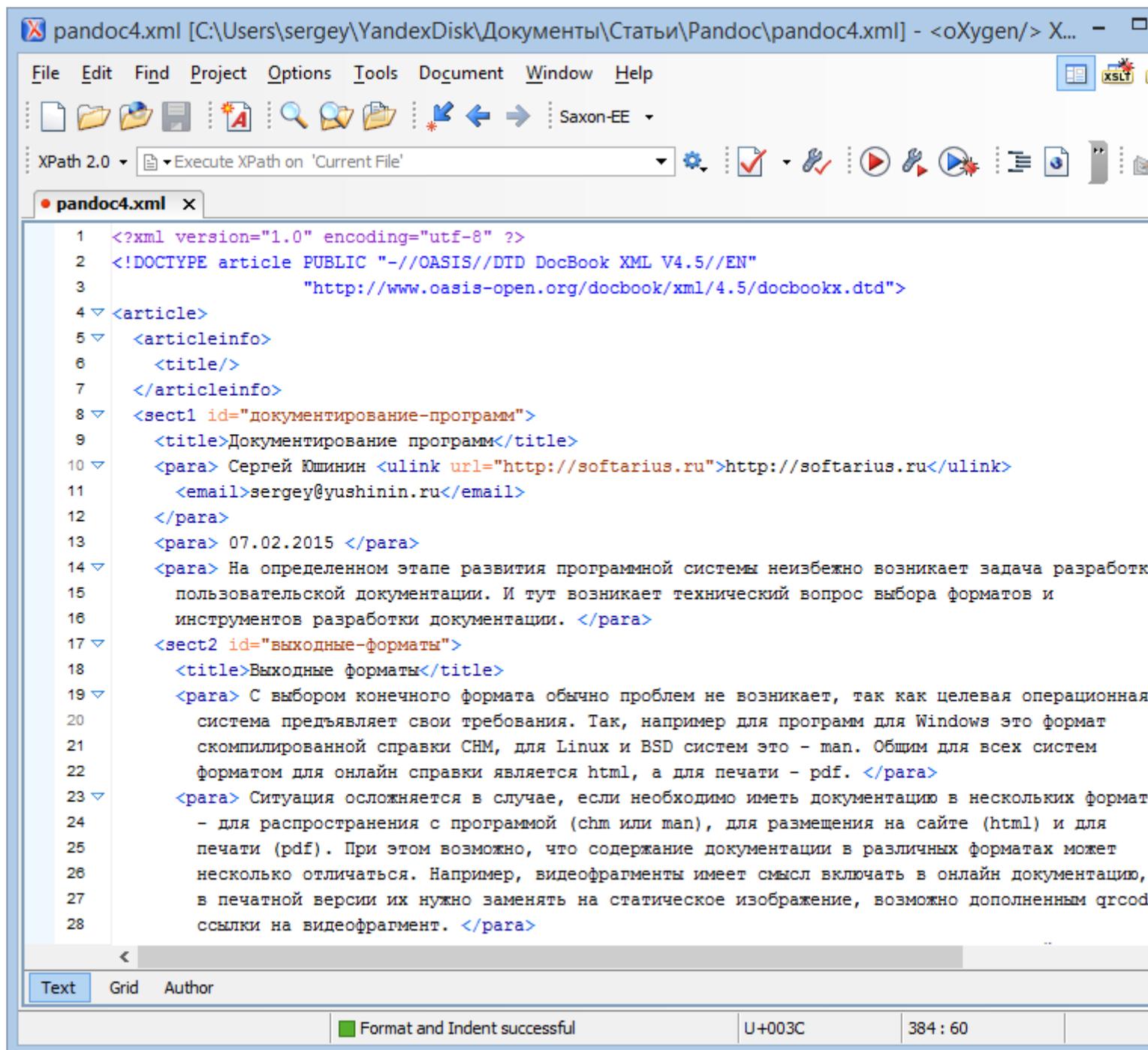


Рис. 8. Сформированная статья в XML-редакторе

Следует отметить несколько дополнительных требований к разметки markdown, которая будет использована для преобразования в docbook:

Во-первых, следует избегать использования в тексте символов угловых скобок (< и >), так как в XML они используются для выделения тегов, а конвертер оставляет их как есть. Если же угловые скобки нужны по смыслу, то следует использовать сущности < и >.

Во-вторых, при вставке рисунка *обязательно* вводить альтернативный текст, так как pandoc использует его для создания обязательного тега title у тега figure.

Однако выходной текст формируется в устаревшем формате Docbook 4 версии, в то время как современная 5 версия предоставляет существенно более богатые возможности по семантической разметке. Для преобразования текста из 4 в 5 версию можно воспользоваться специальным преобразованием `db4-upgrade.xsl`, входящим в комплект поставки Docbook.

```
xsltproc -o pandoc5.xml db4-upgrade.xsl pandoc4.xml
```

Полученный таким образом xml файл схемы docbook 5 можно использовать при формировании единого источника.

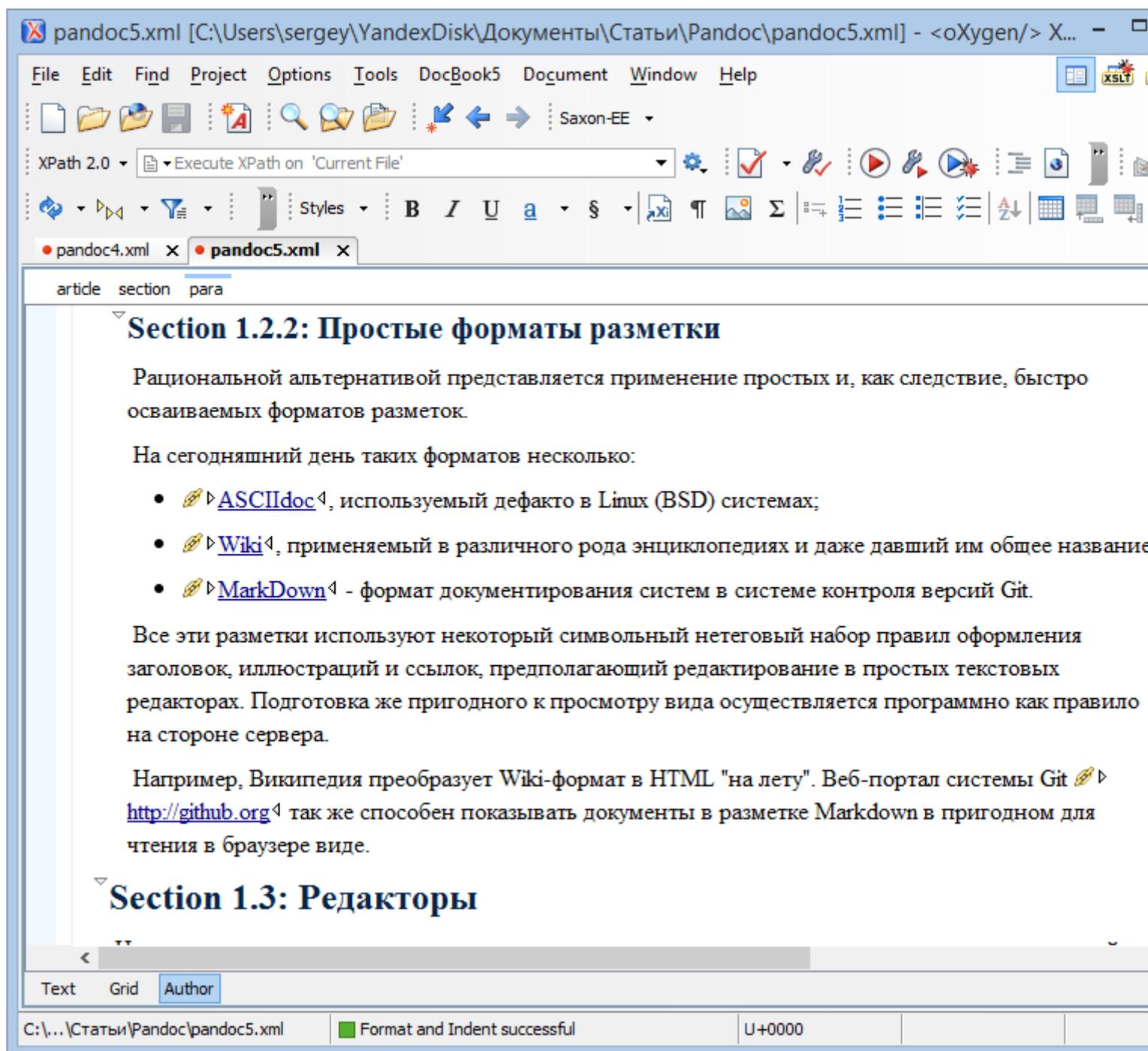


Рис. 9. Статья схемы в XML-редакторе в режиме автора

Описанная цепочка преобразований может показаться на первый взгляд длинной и неоправданно сложной. Однако освоив один раз необходимые инструменты и разработав для часто выполняемых задач командные файлы (скрипты) можно сэкономить значительное количество времени в дальнейшем.

Следует особо подчеркнуть, что технология единого источника обладает ярко выраженным кумулятивным эффектом. Начальные временные затраты на разработку типовых неоднократно используемых фрагментов текста окупаются при их использо-

вании в последующих проектах. Именно это качество делает особо привлекательным технологию единого источника при документировании серийных программных систем.

Набор преобразований Docbook поддерживает формирование документов в HTML со стилями, PDF для печати так сказать "из коробки".

```
xsltproc -o pandoc5.fo c:\<Путь к DocbookXSL>\fo\docbook.xsl pandoc5.xm
```

Внешний вид выходных документов может быть до определенной степени настроен с помощью параметров. Полученные файл формата FO-XSL pandoc5.fo является промежуточным и нужен для построения конечного PDF.

Немаловажна и возможность автоматического формирования оглавления, списка иллюстраций, листингов, таблиц, индексного указателя, глоссария терминов и списка литературы.

При большом количестве документов в составе пакета также возможно создание отдельного списка с возможностью автоматического формирования правильно оформленных ссылок на них.

В случае же подготовки типографского макета руководства с учетом особых требований, например ГОСТ, необходимо разработать дополнительные xsl для форматов обычных страниц, титульной и финальной страницы.

Это может стать темой следующей статьи.

