

Мои заметки о Латехе

В. Г. Кон,

д.ф.м.н., г.н.с. НИЦ КИ, 11-07-2000; 06-01-2011; 23-07-2019

<http://kohnvict.ucoz.ru/main.htm>

Содержание

1. Вместо Введения
2. Инструменты для работы
3. Еще раз о пакетах
4. Литература по Латеху
5. Латех, Интернет и ссылки
6. Новые команды и собственный стиль
7. Расширение возможностей Миктеха
8. Весь документ одним файлом

1. Вместо Введения

Латех существует уже много лет по компьютерному измерению времени, с 1984 года. За это время было написано много самых разных документов по Латеху, в том числе и книг на русском языке. Но как сам Латех бесконечен, так бесконечно и число вариантов его использования. И каждому, как правило, надо что-то свое, особенное, чтобы выразить себя в творчестве издания красивого документа. Я давно и часто пишу документы для себя, чтобы не забыть то, что лично мне нравится и что я массово использую. Увы, это необходимо когда делаешь очень много разных дел, так как запомнить все нехватает памяти. Известно, что еще Эйнштейн говорил, что надо помнить не факты, а место, где про них можно узнать. Самое лучшее место – это свой собственный документ. Хотя я пишу этот текст для себя, но рассчитываю, что и другим будет интересно, поэтому иногда буду описывать понятия, которые мне самому запоминать не надо. Но мой опыт показывает, что появилось новое поколение, люди успешно работают в социальных сетях интернета и даже не знают что такое текстовый редактор, так как нигде ничего не читали и не учили.

Вот для таких людей сообщаю, что Латех – это мощнейшая издательская система, но основанная на принципах программирования, то есть компьютер работает не по принципу (Двое из мешка) или WYSIWYG (без комментариев, кто не знает, смотрим в поиске Яндекса), а по программе. Он сразу делает много работы, а не много раз по разу. WYSIWYG система удобна и наглядна, но она превращает компьютер в лопату, совок в песочнице. Копнул, увидел ямку, еще раз копнул – еще увидел. Но если надо копать 1000 одинаковых ямок, то 1000 раз и придется руками все это делать и каждый раз смотреть. А не проще ли дать команду машине и пусть она работает. Ненаглядно, но быстро. Вот так Латех и работает. Сначала пишем программу документа, потом запускаем компьютер, и он эту программу выполняет. А на выходе мы получаем сразу готовый документ. Все хорошо, кроме одного. Человек слаб, он не видит всего сразу и делает ошибки. Ну что же, ошибки можно исправить в программе документа и потом придется повторить работу. Чем опытнее человек, тем

меньше ошибок. В Латехе есть очень много средств для того, чтобы свести ошибки к минимуму.

Немного скажу об этом. Латех имеет некий стандартный набор команд, точнее оболочек форматирования. Эти оболочки сами являются процедурами для другого языка Тех, у которого уже набор команд фиксирован, но очень большой. В принципе, для Теха можно написать другой набор процедур, не Латех, а скажем Мойтех. Но Латех уже написан, является стандартом, и лучше с ним и работать. Однако и в Латехе можно организовать новые, свои собственные команды, причем неограниченно много. То есть набор процедур Мойтех можно сделать как расширение Латеха. Если это действительно сделать, то ошибок будет меньше. Детали будут потом, пока только общие принципы.

У Латеха есть такое понятие как стили оформления документа в целом и готовые наборы процедур, которые пользователь даже не смотрит, но компьютер смотрит. Такие наборы воспринимаются как черные ящики, про которые надо знать что на входе и что на выходе. А как это работает, знать не надо. Каждый документ имеет всего один стиль, который называется классом и записывается в файл с расширением `cls`. Название этого класса указывается в первой строке программы документа, которую я дальше буду называть `tex`-файл. Поговорим о классах. Они задают стиль оформления документа в целом. Но, в принципе, вас никто не ограничивает следовать этому стилю. Вы можете сделать документ таким, что у него будет другой стиль. По этой причине, если вы создаете документ для своих собственных нужд, то вполне достаточно использовать класс `{article}`, который всегда имеется в любой программе компиляции `tex`-файлов. Совсем другое дело, если вы готовите документ для какого-то издательства. Каждое издательство разрабатывает свой стиль и бесплатно предлагает файл с расширением `cls`, а также инструкцию, как его использовать. Тут уж надо строго следовать инструкции, и вы сразу получите документ, выполненный в стиле этого издательства.

Итак, с классами определились. Еще есть готовые наборы процедур, которые называются пакетами. Их как раз может быть много, и они подцепляются к документу в строках после первой. В принципе есть и другой способ, но лучше знать меньше, да лучше. Для документов на русском языке с возможностями писать цветной текст, вставлять рисунки и делать гиперссылки стандартный набор пакетов, то есть первые строчки `tex`-файла должны быть такими

```
\documentclass[12pt,a4paper]{article}
\usepackage[T2A]{fontenc}
\usepackage[cp1251]{inputenc}
\usepackage[english,russian]{babel}
\usepackage{graphicx}
\usepackage{color}
\usepackage{wrapfig}
\usepackage[colorlinks,urlcolor=red]{hyperref}
```

Следующая проблема – в стиле `{article}` фиксируются размеры области текста на странице, размеры полей и так далее. Но это можно варьировать, никто не заставляет

строго следовать этому стилю, я уже сказал об этом. Если вы хотите задать свои размеры области текста и полей, то ставим в tex-файле через пустую строку вот такой набор команд

```
\oddsidemargin=0mm
\evensidemargin\oddsidemargin
\textwidth=160mm
\textheight=238mm
\topmargin=-14mm
\parindent=0mm
\parskip=3mm
```

Это уже мой стиль – стиль этого документа. Что здесь написано: поле на нечетной странице стандартное, поле на четной странице равно полю на нечетной, знака равенства нет, уже не помню почему, всю жизнь так работаю. Ширина текста 160 мм, высота 238 мм, верхнее поле на 14 мм меньше стандартного. А еще две команды задают нулевым отступ текста в начале параграфа (нулевой абзац), но зато параграфы отделены друг от друга вертикальным пространством 3 мм. Я так люблю и я так сделал, а кто хочет по другому – может сделать по другому.

Указанный выше набор команд является Преамбулой, это информация для компьютера – как форматировать весь текст в целом. Собственно документ у нас еще пустой, в нем нет ни одной буквы. Но прежде чем перейти к документу, поговорим об инструментах.

2. Инструменты для работы

Что нам надо иметь. Во-первых, текстовый редактор для записи исходного tex-файла. Текстовый редактор – это простая программа, которая записывает текст путем набора с клавиатуры, но без форматирования. Таких редакторов очень много и самых разных. Но так как нам в tex-файле надо набить не только текст, но и команды, а потом совершать какие-то специальные действия, то лучше всего использовать специализированный редактор, предназначенный для Латеха. На сегодня, пожалуй, самым лучшим редактором является программа TeXnicCenter, которую бесплатно можно скачать с сайта

<http://texniccenter.org>.

Лучше всего эту программу устанавливать не первой, а после установки программы компиляции tex-файла в документ. Для пользователей ОС Виндовс самая лучшая программа компиляции – это MikTeX. Она тоже бесплатная и у нее было много версий. Есть программа в переносимом варианте (portable), которую не надо устанавливать. Это особенно удобно для тех, кто иногда работает на чужих компьютерах. На чужом компьютере не придется ничего устанавливать. Кроме того, эта программа устанавливается намного быстрее, чем полная версия, просто копируются нужные файлы и все дела. Скачать программу можно по адресу

<http://www.miktex.org>,

там надо выбрать ссылку на MikTeX portable и скачать, как обычно.

Исходно программа не настроена на какой-либо язык, хотя может работать с многими языками. Поэтому после разархивации дистрибутива надо запустить единственный файл с расширением `cmd`. Появится иконка Миктеха в правом нижнем углу экрана компьютера. Ее надо активировать правой кнопкой мыши и выбрать меню [MikTeX Options]. Далее действия по закладкам:

"Tex Formats"

Выделить только `pdflatex`, `tex`, `latex`, `pdftex`

Все остальные форматы "exclude". Нажать "Apply" У меня так и было сразу.

"Languages"

Использовать переносы только для следующих языков:

`english`, `russian`, `dumylang`, `nohyphen`. Нажать "Apply"

"Packages"

Languages/Cyrillic

[Cyrillic, Cyrillic hyphenation pattern, lh, lhcyrillic, ruhyphen, t2]. Нажать "Apply".

"General"

Нажать "Refresh FNDB", затем "Update Formats"

На этом предварительная установка Миктеха завершена и можно компилировать документы на русском языке с переносом русских слов. Интересно, что если провести такую установку на каком либо компьютере, то потом можно просто копировать папку программы на любой другой компьютер и в новом месте все установки сразу заработают. То есть установки делаются не в ОС (операционная система), в самих файлах программы. Правда архив имеет размер 100 Мб, а сама папка программы - 300 Мб. Но ее тоже можно заархивировать при переносе.

Так вот, если сначала установить Миктех, а потом ТехникЦентр, то вторая программа сразу узнает по каким командам надо компилировать `tex`-файл и потом смотреть PDF файл. Компиляция – это процесс, в результате которого программа Миктех читает Латех файл и готовит PDF файл. То есть из программы документа делает полноценный документ. Но если вы это сделаете наоборот, то соответствующие команды можно самому указать по запросу. Такая ситуация возникает, когда меняют одну версию Миктеха на другую. Интересно, что программа ТехникЦентр позволяет работать таким образом, что о наличии программы Миктех вы даже можете забыть. Она сама ее использует. Однако в наборе программы Миктех есть и такие операции, которые ТехникЦентр напрямую не использует. Например, можно конвертировать `eps` формат документа в `pdf`, можно конвертировать `eps` файл рисунка в `pdf`. Для тех, кто не знает, что такое `eps`, `ps`, `pdf`, предлагаю посетить мой сайт

<http://postscript-vk.narod.ru>.

Там об этом написано все и даже больше.

Небольшое полезное замечание. Редактор программы ТехникЦентр исходно запускается без номеров строк, а программа Миктех указывает на ошибки в `tex`-файле именно по номерам строк. Поэтому удобно в редакторе ТехникЦентр сразу выставить режим номеров строк. Для установки этого режима нужно выбрать меню Tools → Options → Editor и отметить галочку у строки ShowLineNumbers.

В принципе это все, и очень многие люди так и работают. Но есть одно но. Как известно, любой научный документ, а Латех как раз наиболее эффективен для научных документов, состоит из четырех компонентов: текста, формул, рисунков и таблиц. Что касается текста, то он и в Африке текст, конечно нужно добавить команды разметки, но их не так много. Рисунков тоже немного и их легко вставить. С таблицами можно создать свою вспомогательную программу. А вот формулы содержат символы, которых нет на клавиатуре. Такие символы набираются кликанием в таблицах иконок, а кодируются длинными словами. Запомнить все эти коды очень сложно. И тут как раз на помощь приходит ТехникЦентр, он позволяет набирать формулы, кликая на иконках математических символов и команд. При этом редактор сам впечатывает длинные команды, которыми кодируются эти символы. Это ускоряет процесс, но не делает его более наглядным, а потому трудно сразу заметить ошибку.

Так вот, я хочу заметить, что существуют программы, которые упрощают набор формул в Латехе. Более того, это полноценные издательские системы, которые способны создавать документ любой степени сложности, но они используют запись форматирования в виде команд Латеха. Первой такой программой была программа, которая называется Scientific Work Place или короче SWP. Эта программа является полностью самостоятельной издательской системой, плюс она умеет вычислять и строить графики. Сложность только в том, что она не бесплатная. Вот сайт этой программы <http://www.mackichan.com>.

Версии программы медленно, но все же растут, а также есть разные варианты программы: Полная система SWP, Scientific Word и Scientific Notebook. Как я понимаю, Scientific Notebook – самая дешевая и ее вполне хватит. Это текстовый редактор, который умеет набирать формулы в манере WYSIWYG, то есть как в Ворде при набивке формул по иконкам вы сразу видите формулы, причем формулы набираются намного удобнее, чем в Ворде, быстро и легко копируются. А спасается документ командами Латеха. Программа умеет также форматировать документ в pdf формат и распечатывать его. Но это уже излишне.

Спасенный текст с командами Латеха можно сразу скопировать в tex-файл программы TeXnicCenter и дальше работать самостоятельно с помощью программы Миктех. В принципе программа даже пишет русский текст с формулами, но русский текст она кодирует не ASCII кодами в Виндовс кодировке, как их использует МикТех, а своим собственным способом. Поэтому нужна дополнительная программа перекодировщик. Я для себя такую программу давно написал и активно ей пользуюсь. Могу бесплатно выслать ее любому по запросу. А если будет много желающих, то и в интернет могу выставить. Итак SWP или Scientific Notebook (часть SWP) очень удобна для набивки формул. Есть три способа получить программу: скачать пробную версию, ее дают на месяц бесплатно, скачать в интернете программу с кей-генератором, попросить у друзей, у кого она уже есть. Еще раз повторяю, можно и без нее обойтись, но для набивки формул она очень удобна. А вот нестандартные команды форматирования текста она не понимает. И рисунки вставляет по своему, то есть нестандартно. Важно, что для подготовки статей в издательстве с нестандартными классами она не годится в полном объеме. Самый лучший способ - это комбинировать ее с программой TeXnicCenter.

Относительно недавно появилась другая программа, имеющая некоторые преимуще-

ства по сравнению с SWP. Программа называется LyX. Вот сайт этой программы:
<http://www.lyx.org>.

Первым ее преимуществом является то, что она бесплатная. Вторым преимуществом является ее способность работать во всех операционных системах, а SWP работает только в Виндовс. Программа тоже самодостаточная, то есть она готовит документ от начала до конца в системе WYSIWIG, но она все же использует Миктех. Хотя ее редактор и не такой красивый, а стандартно она спасает файлы не в формате Латех, а в своем формате, но можно ее попросить и она спасет файл в формате Латех. Более того, она даже умеет спасать русский текст аски-кодами, если ее попросить. Что же касается записи формул, то это происходит примерно таким же образом, как и в программе SWP. То есть, если у вас нет SWP, вполне можно обойтись с помощью этой программы, а через какое-то время вам покажется, что лучше ее вообще ничего нет. Во-всяком случае она все умеет, а к остальному можно привыкнуть. Я не буду подробно объяснять как с ней работать, у нее есть достаточно подробное описание. Важно, что сначала надо установить Миктех и только потом можно устанавливать LyX.

Есть и другие программы для записи tex файла исходного документа в Латехе. Некоторые из них описаны в статье на моем сайте по адресу
<http://kohnvict.ucoz.ru/Data/prolink.htm>.

Хотя и там не полный список. Также я написал программу на своем языке ACL, которая автоматически готовит pdf файл сразу из документа, который спасает программа SWP. То есть она сама перекодирует символы текста, предлагает запустить МикТех, и потом выдает pdf файл. Я сам ей часто пользуюсь для простых документов с формулами.

3. Еще раз о пакетах

Я уже говорил, что пакеты расширяют возможности стандартного Латеха. С их помощью можно делать нечто такое, что выходит за рамки первоначальной идеи форматирования текста и формул. Рассмотрим указанные выше пакеты. Первые три `{fontenc}`, `{inputenc}`, `{babel}`

необходимы для распознавания и обработки русских текстов. В документах на английском языке они не обязательны. Но данные пакеты только указывают, что в документе будет русский текст, однако правила переноса русских слов они не задают. В результате, русские слова будут переноситься целиком и часто будут вылезать за правый край, так как Латех не любит делать большие промежутки между словами. Как раз настройку программы на перенос русских слов надо делать отдельно и об этом было написано в предыдущем разделе.

Пакет `{graphicx}` – это файл `graphicx.sty`, который входит в стандартный дистрибутив Миктеха. Для всех таких файлов есть описание как ими пользоваться в папке `[doc/latex]` внутри главной папки Миктеха. Там есть папка `[graphics]` и в ней файл `graphicx.pdf`. В этом файле описан весь tex код пакета и инструкция - какие команды надо вставлять в текст. Главной командой является такая

```
\includegraphics[width=30mm, angle=0]{picture.pdf}
```

Эта команда вставит картинку из файла `picture.pdf` на текущее место в тексте и

автоматически промасштабирует ее на ширину 30 мм с сохранением аспектного отношения. Вместо ширины можно указать высоту `height=30mm` или масштабирующий множитель `scale=2`. Можно также повернуть картинку на любой угол `angle=90` и даже указать точку вокруг которой будет осуществляться вращение `origin=c` или `origin=lb`. Здесь буквы означают l-left (лево), r- right (право), b-bottom (низ), t-top (верх), c-center (центр). Можно указать и конкретную точку с координатами x,y. Наконец, можно указать слово `clip` и тогда картинка будет не промасштабирована, а будет вырезана из оригинала в окно заданного размера. В этом случае возможно надо задавать параметр `bb = x y w h`, то есть координаты вырезаемого прямоугольника. Я сам ни разу этим не пользовался. Есть и другие возможности, но они редко используются.

Я замечу, что аргументы практически любой команды делятся на обязательные и необязательные. Обязательные помещаются в фигурные скобки, а необязательные – в квадратные скобки. Таким образом, в данной команде обязательным аргументом является имя файла картинки `{picture.pdf}`. Этот файл должен присутствовать на компьютере в текущей папке, то есть в той же папке, где и `tex`-файл. При этом принцип такой. Если вы предполагаете компилировать из Латеха сразу в `pdf`, то картинка должна быть в `pdf` формате, так как она просто подцепляется на стадии компиляции без конвертирования форматов. Если же вы намерены конвертировать в `dvi` формат, то картинка должна иметь `eps` формат. При последующем конвертировании `dvi` в `ps` и затем `ps` в `pdf` она преобразуется вместе с текстом в `pdf` формат.

Конвертировать `eps` формат в `pdf` можно программой Миктеха `epstopdf.exe`, которая находится вместе со всеми другими программами Миктеха в папке `miktex\bin` внутри главной папки Миктеха. Имя `eps`-файла для конвертирования надо поставить аргументом к этой программе при вызове ее с командной строки. Проще всего сделать `bat` файл с таким содержанием

```
set path=c:\_vk\Miktex\miktex\bin;%path%
epstopdf.exe имя-файла
```

и запускать его простым кликом. Здесь указан путь к файлу как он стоит у меня на компьютере. Если у вас по другому, то надо поменять путь, а также вместо слов имя-файла надо поставить реальное имя файла. С помощью таких `bat` файлов можно запускать большое число программ Миктеха, в том числе программу конвертирования `ps`-документа с многими страницами и рисунками в `pdf`-формат (`ps2pdf.exe`), программу конвертирования Латеха в `PDF` (`pdflatex.exe`) и многое другое.

Все параметры, которые стоят в квадратных скобках, необязательны. Можно ничего не ставить. Но тогда команда будет делать вариант по умолчанию. Необязательные параметры позволяют регулировать этот процесс. Если указателей много, то они разделяются запятой и пробелом.

Данный пакет определяет еще одну команду, а именно

```
\rotatebox[origin=c]{180}{text}
```

Эта команда может вращать текст вокруг указанной точки на заданный угол. Хотя эта команда редко используется, но была бы возможность, а применение найдется.

Описанный выше пакет является основным для вставки рисунков в документ. Разу-

меется были и другие пакеты, и иногда в руководствах они упоминаются. Но данный пакет считается наиболее развитым и с 1999 года ничего другого не было предложено. Вероятно это как раз и показывает, что в нем есть все, что нужно.

Следующий пакет `{color}` – это файл `color.sty`, он тоже входит в стандартный дистрибутив Миктеха. Соответственно есть и описание. Этот пакет позволяет вводить цвет в документ. Ниже я перечислю основные команды этого пакета

`\color{blue}` – переключает цвет последующего текста на синий. Использует предопределенный цвет. Всего предопределенных цветов не так много, они следующие `black, white, red, green, blue, cyan, magenta, yellow`

Для повторного изменения цвета снова используем эту команду.

`\textcolor{red}{text}` – эта команда меняет цвет только у текста, который стоит ее вторым аргументом. Она удобна для выделения отдельных слов.

`\definecolor{mycol}{rgb}{0.9,0.8,0.7}` – эта команда определяет цвет как совокупность красного, зеленого и синего и присваивает ему имя. В дальнейшем этот цвет можно задавать по имени.

`\colorbox{mycol}{text}` – эта команда задает цвет страницы под текстом (фона). Цвет текста не меняется, но выделяется цветом прямоугольник, в котором он находится.

`\fcolorbox{green}{red}{text}` – эта команда помещает текст в рамку и использует два цвета. Первый цвет задает цвет рамки, второй цвет задает цвет фона.

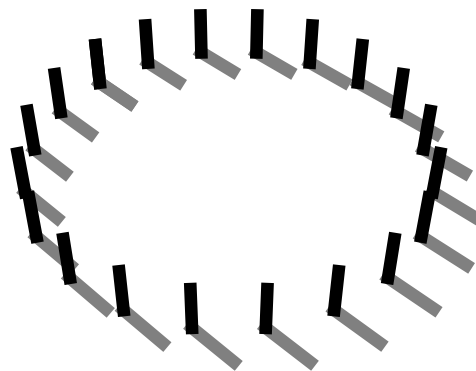
`\pagecolor{yellow}` – эта команда задает цвет данной страницы и всех последующих страниц.

Пакет `{wrapfig}` – это файл `wrapfig.sty`, но он не входит в стандартный дистрибутив Миктеха. Этот пакет написан одним из пользователей Латеха и предназначен он для более фасонной работы, а именно, вставки рисунка, обтекаемого текстом. Дело в том, что описанный выше пакет `{graphicx}` вставляет рисунок в текст на место следующей буквы. Если высота рисунка большая, то строка соответственно увеличивается в размере и больше ничего не происходит. Это не всегда удобно, поэтому был разработан данный пакет. Где взять этот пакет? В интернете существует огромная библиотека пакетов для Латеха по адресу

<http://www.ctan.org>.

Достаточно туда зайти, указать какой файл вам нужен и вам сразу дадут ссылку для скачивания файла. А что с описанием? Оно находится в самом файле после кода.

Данный файл описывает новое окружение. Если поместить команду вставки рисунка в это окружение, то вставка рисунка происходит по другому, а именно, последующий текст автоматически обтекает рисунок. Чтобы не быть голословным я сразу показал как это выглядит в документе. Справа вы видите рисунок, который был поставлен с использованием окружения данного пакета. Рисунок проще всего ставить в начале



следующего параграфа. Рисунок вставляется сам по себе, а последующий за ним текст обтекает его автоматически. То есть все очень просто.

Код вставки рисунка, использованный в данном документе, выглядит следующим образом

```
\begin{wrapfigure}{r}{75mm}
\includegraphics[width=70mm]{exam-07.pdf}
\end{wrapfigure}
```

Сразу за ним идет текст обтекаемого параграфа. Как вы видите, команда окружения имеет два аргумента. Первый указывает сторону (r-right справа и l-left слева), второй указывает ширину области, которая будет отведена для рисунка. Я указал ширину области 75 мм и промасштабировал рисунок на ширину 70 мм. Так как мой рисунок векторный, то он масштабируется точно. Для растровых рисунков нужно подбирать размер так, чтобы при показе pdf файла с размером 100% пиксели рисунка соответствовали пикселям экрана. Но если даже этого и не произойдет, то рисунок все равно промасштабируется и будет выглядеть нормально.

Стандартный режим хорошо работает, когда обтекаемый текст не содержит формул, таблиц и прочих нестандартных объектов. Если же это имеет место, то могут быть ошибки в подсчете обрезаемых строк. Для этих целей существует полная версия окружения

```
\begin{wrapfigure}[12]{r}[34pt]{75mm}
\includegraphics[width=70mm]{exam-07.pdf}
\end{wrapfigure}
```

Первый необязательный параметр (в квадратных скобках) явно указывает число обрезаемых строк, второй явно указывает вертикальное пространство выше и ниже рисунка. Стандартно это вертикальное пространство равно примерно высоте одной строки. Это не всегда красиво выглядит.

4. Литература по Латеху

Разумеется есть другие варианты установки рисунка в документе. Есть стандартное окружение `\begin{figure}`, которое позволяет вставить рисунок с подписью. Внутри него можно использовать среду `\begin{minipage}`. Я не стану здесь это описывать, это все можно найти в стандартных описаниях Латеха, которых много в интернете. В частности можно скачать книгу Львовского на русском языке, кликая на ссылке

<http://www.mccme.ru/free-books/llang/newllang.pdf>

Это 3-е издание 2003 года и там много всего написано про Латех.

Есть также перевод на русский язык известного английского руководства по Латеху

<http://www.ccas.ru/voron/download/books/tex/oetiker99latex.pdf>

Кроме того, существует справочник по всем командам Латеха. Его можно скачать кликая на этой ссылке

<http://www.ccas.ru/voron/download/books/tex/klimenko96sprav.pdf>

В той же папке находится просканированная книга про Латех, довольно старое издание, но очень полное и тоже на русском языке

<http://www.ccas.ru/voron/download/books/tex/gussens99latex.djv>

Весьма полезным может оказаться справочник конструкций Латеха с примерами.

Он в краткой форме содержит все команды Латеха и очень хорош для тех, кто уже все знает, но не помнит. То есть для всех.

<http://www.ccas.ru/voron/download/voron05latex.pdf>

5. Латех, Интернет и ссылки

Хотя Латех и старая идея, но она развивается. В частности в последние годы получила развитие техника, при которой pdf-файлы содержат кликабельные ссылки на интернет ресурсы. Клик на такой ссылке автоматически запускает браузер по умолчанию, который показывает интернет ресурс по данному адресу. Таким образом, pdf документы оказываются интегрированными в интернет. Как это делается в Латехе? Прежде всего в Преамбуле нужно заказать пакет, который как раз и реализует такую возможность. Наиболее общая форма заказа пакета выглядит так

```
\usepackage[colorlinks,urlcolor=red]{hyperref}
```

Здесь сразу заказаны цветные ссылки и в качестве цвета выбран красный. Если пакет заказан, то вставить ссылку в тексте документа можно с помощью команды

```
\href{URL-address}{text}
```

Команда имеет два обязательных аргумента. Первый задает полный URL адрес, начинающийся с `http://`, а второй указывает текст, который будет реально виден в документе для этой ссылки. При клике на этот текст будет показан сайт по указанному адресу.

Есть еще и другая команда с одним аргументом

```
\url{URL-address}
```

в которой указывается URL адрес, но без `http://` и именно этот адрес и показывается. Но при клике на него к адресу добавляется `http://` и он работает как предыдущий адрес. Ниже будет показано как свести первую команду к второй.

Таким образом все просто. Конечно в данном вопросе pdf файлы копируют функции html файлов, но у них есть возможность создавать документ с формулами и рисунками и причем в виде отдельного файла, в то время как html файлы при их использовании без интернета заметно проигрывают.

Для более продвинутых можно указать еще дополнительные возможности. Команда

```
\hyperbaseurl{URL-address}
```

задает базовую часть гиперссылки, после чего все остальные ссылки относятся к этой, то есть достаточно указать остаток ссылки после вычитания данной базовой части. Команда

```
\hyperimage{URL-image}{text}
```

используется для ссылки на картинки по их интернет адресам.

Далее, команда

```
\hyperdef{category}{name}{text}
```

присваивает тексту адрес, то есть ставит метку, которая имеет вид `category.name`.

А другая команда

```
\hyperref{URL}{category}{name}{text}
```

работает как ссылка на адрес `URL#category.name`. Довольно интересная команда

```
\hyperref[label]{text}
```

делает текст ссылкой на то место документа, на которое указывает ссылка `\ref{label}`. Для ссылок внутри документа наиболее удобна пара команд

```
\hypertarget{name}{text}
\hyperlink{name}{text}
```

Первая из них присваивает тексту имя, а вторая делает ссылку на это имя. Такие ссылки внутри документа можно использовать и без интернета.

В действительности здесь указаны не все команды, есть и другие. Но указанных команд может хватить на длительное время, после чего можно просто почитать документацию. Читать ее непросто, но если будет охота, то ...

6. Новые команды и собственный стиль

Латех – это переопределенная система, в ней один и тот же результат можно получить разными способами. С одной стороны, общий вид документа задается классом, который описан в файле с расширением `cls`. Этот класс определяет полный набор команд, которые задают стиль документа. С другой стороны, вы вовсе не обязаны использовать все эти команды. Если вы посылаете статью в издательство, то тогда действительно надо следовать инструкции, поскольку в издательстве вполне возможно есть программы автоматического набора, использующие стандартные команды. Но если конечной целью является создание pdf файла документа, то вы можете проигнорировать рекомендуемые команды и вместо них форматировать как размеры текста на странице, так и общее оформление своим способом. Более того, вы можете создать некоторый набор своих собственных команд, которые будут автоматически использоваться по записи ваших новых названий.

Этот набор новых команд может быть небольшим и помещаться в каждый файл документа, а можно записать его в отдельный файл и просто импортировать в файл документа. Определение новых команд – это очень большое преимущество всех интерпретируемых языков программирования. При этом возникает некоторое уменьшение автоматизма, но все в разумных пределах. Так если вы используете стандартные команды

```
\title, \author, \section, \subsection
```

то Латех способен автоматически составить оглавление, автоматически пронумеровать разделы и так далее. Но это как раз несложная работа и в ней от автоматике можно отказаться. И вместо этих команд определить свои

```
\newcommand\Tit[1]{\textcolor{blue}{\Large\bf #1}}
```

 для заголовка статьи

```
\newcommand\Sec[1]{\textcolor{blue}{\large\bf #1}}
```

 для названия раздела

При этом стиль вашего документа будет не совпадать со стандартом и может быть произвольным при использовании любого класса. Определение новых команд может находиться где угодно в документе, но лучше всего их поставить в преамбуле. Сама команда называется `\newcommand`, она имеет два обязательных аргумента: первый указывает название новой команды, второй – ее определение, то есть тот текст, который будет проставлен в документе вместо названия. Вообще говоря, обязательные аргументы надо ставить в фигурные скобки, но у меня первый аргумент записан без фигурных скобок и это тоже работает, вероятно так было в старых версиях Латеха и осталось для совместимости.

Новая команда тоже может иметь аргументы, это ее переменная часть. Число аргументов ставится в квадратных скобках, а сами аргументы обозначаются конструкцией типа #1, #2 и так далее. Так, например, команда `\Tit{ }` имеет один аргумент, название статьи документа. Это название пишется с определенным оформлением, которое уже зафиксировано в определении команды и не может изменяться. Меняться может только само название. То же самое с командой `\Sec{ }`. Она очень компактно записывается и может использоваться многократно. Что касается автора, то его можно записать и без команды. При таком подходе номера разделов надо писать руками и никакой автоматике не будет. Но их ведь не так много.

Следующие три команды сделаны для гиперссылок как на интернет, так и внутри документа

```
\newcommand\mhr[1]{\href{#1}{\bf #1}}
\newcommand\hn[1]{\hypertarget{#1}{\ \ \ }}
\newcommand\hl[2]{\hyperlink{#1}{\bf #2}}
```

Особенностью команды `\mhr{ }` является то, что она понижает число аргументов. Если исходная команда `\href{ }{ }` имела два аргумента, то новая – всего один, который два раза повторяется. При этом указывается полная интернет ссылка и она же является кликабельной. Команда `\hn{ }` также понижает число аргументов, но другим способом. Она ставит метку на стандартный текст, состоящий из трех пробелов. Фактически при размещении метки сам текст не имеет значения, важно только место, где он находится. А третья команда `\hl{ }{ }` почти ничего не делает, она только принуждает писать второй аргумент жирным шрифтом. Казалось бы пустяк, а можно забыть, да и неинтересно много раз писать одно и то же. Ну и она сокращает название команды. Эти команды позволяют удобно расставлять ссылки на интернет и названия разделов внутри документа. Хотя оглавление данной статьи и написано вручную, но оно содержит навигацию по документу, что намного удобнее для пользователя.

Напоследок скажу, что новых команд может быть много и можно разметить документ с их помощью таким образом, что он совсем не будет похож на стандартный Латех. Точнее, он будет представлять ваш собственный диалект Латеха. А все ваши новые команды можно записать в отдельный файл и вставлять в документ с помощью команды `\input{file}`. Фактически это является аналогом нового пакета.

7. Расширение возможностей Миктеха

Речь в этой главе пойдет о рекомендованной программе portable MikTeX. Эта программа отличается тем, что она не только переносимая, но еще и компактная. Ее дистрибутив имеет размер 100 Мб, а полностью развернутая программа имеет размер 300 Мб. При этом программа содержит некий минимум разработанных пакетов Латеха, но далеко не все. Я уже упоминал, что в ней нет пакета `{wrapfig}`. И я описал как скачивать пакет из интернета. На самом деле в самой программе есть очень удобный способ подключения недостающих пакетов. Делается это следующим образом.

Запускаем файл с расширением `cmd` (см выше) для того, чтобы сгенерировать икон-

ку Миктеха в правом нижнем углу экрана. Кликаем эту иконку правой кнопкой мыши. В появившемся меню выбираем строку `Miktex Package Manager` и кликаем ее. Через небольшое время появляется таблица всех пакетов Латеха, известных в интернете. Все что нужно сделать – это отметить нужный пакет кликом левой кнопкой мыши, при этом загорается иконка со знаком плюсикиа [+]. Кликаем на плюсики и соглашаемся с установкой данного пакета. Программа автоматически найдет данный пакет в интернете и установит его. Таким способом без труда можно установить пакет `{wrapfig}` и даже не придется его искать.

Вся проблема только в том, что вы сами должны знать какие пакеты вам нужны для тех или иных целей. Информацию об этом можно получить из книг, из интернета, от коллег и так далее. Так, например, для отправки научной статьи в журналы американского физического общества и некоторые другие нужно иметь пакет `{revtex}`. Его легко установить по описанной выше схеме. Но оказывается этого пакета недостаточно. Вместе с ним надо установить пакеты `{natbib}` и `{textcase}`, а возможно и еще какие-нибудь, это может зависеть от структуры статьи. По крайней мере я чисто опытным путем установил, что для моей статьи этих пакетов достаточно.

Сложность в том, что сама программа Миктеха подсказки о недостающих пакетах не делает. Она просто обрывает свою работу как только не находит недостающий пакет. Вероятно программа используется та же самая, что и для полной версии размером 1.5 Гб, в которой есть все пакеты. И поэтому механизм подсказки не предусмотрен. Один из возможных рецептов решения проблемы - попробовать прокомпилировать статью с другой версией Миктеха и изучив log-файл узнать какие пакеты были загружены. А потом проверить их наличие в вашей версии Миктеха.

8. Весь документ одним файлом

Я уже писал, что Латех может работать с большим числом файлов. Во-первых есть команда `\input{file}`. Во-вторых, команды вставки рисунков также указывают на eps или pdf файлы и их может быть много. И если код записан таким образом, что он указывает на внешние файлы, то казалось бы никак нельзя все записать одним файлом. Но такая возможность все же есть. Ее практически нет смысла использовать для себя, ведь всегда можно сделать zip архив из многих файлов при переносе на другой компьютер. Но вся беда в том, что редакции многих научных журналов требуют закачивать Латех документы одним файлом. Многих, но не всех. Иногда это не трубется.

Но головная боль возникает именно в тех журналах, где это требуют. Тогда на помощь приходит специальная оболочка Латеха

```
\begin{filecontents}{имя файла}
содержание файла
\end{filecontents}
```

Таких оболочек может быть много для разных файлов, и они должны находиться в файле первыми, то есть до основного кода программы Латеха. При этом компилятор

делает следующее: если в рабочей папке, где находится файл с документом, уже есть файл с указанным именем, то он ничего не делает. А если файла нет, то он открывает новый файл с таким именем и переписывает туда содержание файла, указанное внутри оболочки. Однако не просто переписывает, а перед содержанием вставляет свой комментарий (начинающийся с символов %) с указанием даты и файла, который этот файл сгенерировал. Файлы с расширением tex и eps такие комментарии игнорируют, но другие файлы могут быть испорчены. Чтобы программа не записывала комментарий можно использовать оболочку со знаком * после имени оболочки, то есть filecontents*.

Для редакций журналов обычно достаточно записать таким образом eps файлы рисунков, и в этом случае обе формы (с символом * или без него) работают одинаково в том смысле, что получается правильный pdf файл даже когда компиляция делается на сервере журнала. Вероятно создатели Латеха разработали такую форму объединения файлов, чтобы не зависеть от внешних программ. Так, например, zip архив по разному может записываться в разных операционных системах, а текст он всюду одинаков, хоть и приводит к большим размерам файлов. Работать с огромными текстовыми файлами в текстовом редакторе не очень удобно, но можно использовать команды системы копирования многих файлов в один.