

Содержание

5. Оформление текста в целом	2
5.1. Большие проекты	2
5.2. Титульный лист	2
5.2.1. Центрирование текста	3
5.2.2. Пропуск места по вертикали и по горизонтали	3
5.2.3. Горизонтальное отчеркивание	4
5.2.4. Выравнивание по правому краю	4
5.2.5. Министраницы	5
5.2.6. Как закончить титульную страницу	6
5.3. Особые требования к оформлению	6
5.3.1. Меняем вид заголовков	7
6. Создание новых команд	7
6.1. Макроопределения	8
6.1.1. Команды без аргументов	8

5. Оформление текста в целом

5.1. Большие проекты

При работе с большими документами вам может быть удобно разделить входной файл на несколько частей. \LaTeX содержит две команды, которые помогают это делать.

```
\include{файл}
```

Эту команду можно использовать в теле документа, чтобы включить в него содержимое другого файла. Заметьте, что \LaTeX начнет новую страницу, прежде, чем обрабатывать материал, содержащийся в *файле*.

Вторая команда может использоваться только в преамбуле. Она указывает \LaTeX читать только некоторые из включенных (`\include`) файлов.

```
\includeonly{filename,filename,...}
```

После выполнения в преамбуле документа этой команды будут выполняться только те команды `\include`, чьи имена файлов перечислены в аргументе команды `\includeonly`.

Команда `\include` начинает верстку включаемого текста на новой странице. Это удобно при использовании `\includeonly`, потому что границы страниц не будут двигаться, даже когда некоторые включаемые файлы пропускаются. Иногда это неудобно, и в этом случае вы можете использовать команду

```
\input{filename}
```

Она просто включает содержимое указанного файла.

Для быстрой проверки документа используйте пакет `syntonly`. Он заставляет \LaTeX пробежать по документу, проверяя синтаксис и использование команд, но не генерируя выходного файла DVI. \LaTeX в этом режиме работает быстрее, экономя ваше время. Синтаксис очень прост:

```
\usepackage{syntonly}  
\syntonly
```

Когда вы захотите получить выходной файл, просто прокомментируйте вторую строчку, добавив знак процента.

5.2. Титульный лист

Титульная страница отличается от всего остального текста тем, что при ее создании нам необходимо явно задать расположение текста — верхней «шап-

ки», имени и фамилии автора, заголовка работы, всевозможных подзаголовков, сведений о научном руководителе, и, наконец, традиционной информации о времени и месте (город/год) создания работы.

До сих пор не касались методов влияния на физическое размещение текста: в большинстве случаев будет лучше, если мы не будем мешать \LaTeX у верстать документ в соответствии с типографскими нормами. Это, однако, не относится к титульной странице.

5.2.1. Центрирование текста

Начнем с команды `\centerline` и окружения `centering`. Легко догадаться, что они позволяют выводить строки текста, центрируя их на листе. С командой `\centerline{text}` все более-менее понятно: она принимает на вход аргумент, обычно заключаемый в фигурные скобки, и выводит его всегда в одну строку (даже если эта строка не влезет на лист), причем середина строки располагается в середине полосы набора.

Речь идет именно о полосе набора, а не о физическом листе, так что если в нашем документе заданы разные величины для левого и правого полей, при центрировании это окажется учтено.

Окружение `centering` позволяет центрировать текст, состоящий из нескольких строк. Если при этом потребовать набрать с центрированием целый абзац, он будет автоматически разбит по имеющимся пробелам на строки такой длины, чтобы каждая из них уместилась в полосе набора.

Здесь необходимо учитывать эффект, с которым мы уже сталкивались при обсуждении управления размером шрифта и интерлиньяжа. \LaTeX форматирует абзац только тогда, когда этот абзац закончился, то есть встречена пустая строка или команда `\par`. Таким образом, если вы заключите некий текст между командами `\begin{centering}` и `\end{centering}`, не отделив последнюю от текста пустой строкой, эффект может вас несколько удивить: ведь \LaTeX «увидит» конец абзаца уже после того, как окружение закончилось, а значит, и форматировать абзац будет как обычно, то есть без всякого центрирования. Простейший вариант решения — всегда отделять начало и конец окружения от текста пустой строкой.

Если автоматически выбранная \LaTeX ’ом разбивка на строки вас не устроила, можно расставить переводы строк вручную, используя команду `\\`.

5.2.2. Пропуск места по вертикали и по горизонтали

Между элементами оформления титульной страницы обычно вставляются свободные участки. Есть два основных способа задать такой промежуток. Первый способ предполагает явное указание величины промежутка (например, в сантиметрах). Второй позволяет равномерно распределить «вакантное» вертикальное пространство между несколькими указанными местами в тексте.

Сделать промежуток фиксированного размера можно с помощью команды `\vskip`, указав (без пробела) размер в сантиметрах (cm), миллиметрах (mm) или пунктах (pt). Например, `\vskip2cm` создаст пустое пространство в 2 см, `\vskip15mm` или `\vskip1.5cm` — в 15 мм и т.д.

Для равномерного распределения вертикального пространства используется команда `\vfill`. По окончании генерации содержания страницы все оставшееся свободное место вдоль вертикальной оси распределяется поровну между встреченными на странице командами `\vfill`. Таким образом, если вставить `\vfill` в одном месте, то все свободное вертикальное пространство будет вставлено именно в это место. Если использовать две команды, на месте каждой будет ровно половина свободного места, если три — соответственно, треть, и т. д.

Если нужно сделать вертикальные отступы разного размера, можно в одном месте использовать несколько команд `\vfill`. Например, если в одном месте вставить две команды, а в другом три, то и свободное место распределится в отношении 2:3.

Аналогичные возможности предусмотрены и для горизонтальных отступов в строках. Команда `\hspace{длина}` вставляет в текст горизонтальный отступ, заданный параметром длина (следует учесть, что эта команда не сработает в конце строки; если вы с этим столкнулись, попробуйте «звездочную» версию `\hspace*`). Команда `\hfill` задает точку, для которой производится распределение свободного горизонтального пространства; иначе говоря, оставшееся свободным горизонтальное пространство в строке распределяется поровну между командами `\hfill`, аналогично тому, как вертикальное пространство распределяется между командами `\vfill`.

5.2.3. Горизонтальное отчеркивание

Чтобы провести горизонтальную черту (в том числе в местах, предполагающих вписывание от руки), используется команда `\rule[высота]{длина}{толщина}`. Здесь обязательные параметры длина и толщина задают длину и толщину проводимой линии, а необязательный параметр высота указывает, насколько линию следует поднять вверх относительно нижнего края букв в строке. Если это не нужно, параметр следует опустить вместе с квадратными скобками.

Полезной может оказаться и команда `\hrulefill`, по сути аналогичная команде `\hfill` из предыдущего параграфа, с той только разницей, что свободное место в строке она заполняет отчеркиванием.

5.2.4. Выравнивание по правому краю

Часто на титульной странице требуется прижать определенные фрагменты текста к правому краю полосы набора. В диссертации, например, именно так размещается код УДК и стандартная надпись «на правах рукописи»; на ти-

тальной странице дипломной или курсовой работы обычно по правому краю выравнивается информация о научном руководителе, а иногда и об исполнителе работы.

Прижать вправо отдельную строку можно командой
Если же прижать вправо необходимо несколько строк подряд, можно воспользоваться окружением `flushright`: все строки, заключенные между командами `\begin{flushright}` и `\end{flushright}`, будут сдвинуты к правой границе полосы набора. Рекомендуется внутри окружения `flushright` разделить строки явным образом с помощью команды `\\`.

Наконец, можно сформировать сдвинутый вправо абзац или несколько абзацев, воспользовавшись окружением `minipage`, которое будет описано в следующем параграфе.

5.2.5. Министраницы

При верстке титульной страницы и других сложных элементов текста могут пригодиться окружения `minipage` и `boxedminipage`. Эти окружения позволяют сверстать министраницу, то есть прямоугольную область текста, внутри которой текст верстается по тем же правилам, что и для обычной страницы (даже, при необходимости, со сносками), а во внешнем тексте министраница обрабатывается как обычный символ, только очень большой.

Окружение имеет обязательный параметр, задающий ширину министраницы. Параметр указывается в фигурных скобках сразу после команды `\begin{minipage}`. Так, например, если нужно сверстать абзац шириной в 30% полосы набора и прижать его вправо, можно воспользоваться следующим кодом:

```
1 \null\hfill
2 \begin{minipage}{0.3\textwidth}
3 Весь этот текст сформатирован в отдельный абзац
4 и сдвинут вправо к краю страницы.
5 \end{minipage}\\
```

Результат будет примерно таким:

Весь этот текст сформатирован в отдельный абзац и сдвинут вправо к краю страницы.

Если подключить в преамбуле пакет `boxedminipage`, вам станет доступно окружение `boxedminipage`, работающее точно так же, как и `minipage`, только сверстанная страница будет обведена рамкой. Например, следующий код:

Это пример министраницы в рамке

даст примерно такой результат:

Это пример министра-
ницы в рамке

5.2.6. Как закончить титульную страницу

Если никак не сообщить \LaTeX ’у об окончании титульной страницы, то текст, следующий непосредственно за ней, будет выведен на этой же странице, что вряд ли соответствует вашим желаниям.

Заявить об окончании титульной страницы можно двумя способами. Более простой из них — команда `\clearpage`. Встретив эту команду, \LaTeX немедленно выведет все накопленные абзацы, распределив оставшееся вертикальное место между командами `\vfill`, и приступит к верстке следующей страницы. При этом, однако, будет выведен на печать номер страницы, что нежелательно. Избежать этого можно, поместив в код титульной страницы в произвольном месте команду `\thispagestyle{empty}`.

Другой способ — заключить содержимое титульной страницы в окружение `titlepage`. В некотором смысле это более правильно. Следует, однако, учесть, что дальнейшую нумерацию страниц в этом случае \LaTeX продолжит с номера 1, тогда как в соответствии с отечественными стандартами первой страницей считается сама титульная страница (хотя номер на ней, естественно, не ставится). Исправить ситуацию поможет команда `\setcounter{page}{2}`, вставленная сразу после окончания титульной страницы (то есть, в данном случае, после команды `\end{titlepage}`). Эта команда принудительно установит номер текущей страницы в значение 2, что, собственно, и требуется.

5.3. Особые требования к оформлению

Когда ваша рукопись окажется, по крайней мере, на ваш взгляд, полностью готова, вам могут предъявить требования, которые кому-то покажутся простыми, но начинающего верстальщика ставят в тупик. Это особенно касается случаев, когда рукопись предназначена для типографского размножения, но может случиться и при оформлении квалификационной работы. Как, например, убедить \LaTeX ставить точки после номеров секций? Как изменить размер шрифта, которым \LaTeX набирает слово «глава» в заголовках глав? Как заставить \LaTeX центрировать заголовки, а то и вовсе прижимать их к правому краю полосы набора, а не к левому? Как в подписях к таблицам и рисункам заменить двоеточие после номера на точку?

Мы попытаемся ответить на эти вопросы, но прежде хотелось бы дать читателю один небольшой совет: не прибегайте к таким (практически крайним) мерам, пока вас к этому не вынудят. Как показывает опыт, принудительные изменения стиля оформления не идут на пользу общему виду документа.

5.3.1. Меняем вид заголовков

Чаще всего нам приходится задуматься о внешнем виде заголовков глав и секций, когда кто-то требует от нас поставить точки после номеров секций и подсекций. В разных источниках, посвященных \LaTeX ’у, можно встретить различные рекомендации, как этого достичь. Чаще всего рекомендуют переопределить команды `\thesection` и `\thesubsection`, однако результат может вас обескуражить: например, при этом «сломается» внешний вид ссылок, заданных командой `\ref`.

Вторая часто возникающая проблема — как «отучить» \LaTeX от привычки переносить слова в длинных заголовках. Добиться этого можно было бы явным запретом переноса соответствующих слов с помощью команды `\mbox`, но при этом мы рискуем увидеть в заголовках чрезмерно растянутые пробелы.

Как правило, на такие изменения в оформлении разработаны отдельные пакеты. Например, если нужно добавить точку после номера в заголовках, нужно воспользоваться пакетом `titlesec` и изменить команду `\titlelabel`

```
\usepackage{titlesec}
\titlelabel{\thetitle.~}
```

Если нужно изменить не только номер заголовка, но и его оформление, нужно переопределить параметр `\titleformat`. Например, следующий пример добавляет точку после номера и запрещает переносы в заголовках 3-го уровня

```
\titleformat{\subsubsection}{\filright\bf}{\thesubsubsection.}{0.5em}{}
```

Если нужно сделать подчиненную разделу нумерацию уравнений, таблиц и рисунков, нужно задать в преамбуле следующие команды:

```
\numberwithin{equation}{section}
\numberwithin{table}{section}
\numberwithin{figure}{section}
```

соответственно.

6. Создание новых команд

Средства \LaTeX ’а, описываемые в этом разделе, позволяют значительно сократить число нажатий на клавиши при наборе сложных текстов. Мы расскажем, как создавать новые команды, заменяющие собой длинные фрагменты из текста и \TeX ’овских команд. Официально такие новые команды называются макроопределениями, а в разговорной речи — макросами.

Начнем с примера. Пусть вы пишете текст, в котором регулярно встречается математический значок $\stackrel{\text{def}}{=}$ (он означает «равно по определению»). Этот значок генерируется внутри математической формулы такой последовательностью команд:

```
\stackrel{\mathrm{def}}{=}
```

Часто писать такой длинный набор команд утомительно. Вот бы в \LaTeX ’е была предусмотрена команда, скажем, (`eqdef`), генерирующая символ бинарного отношения $\stackrel{\text{def}}{=}$! Правда, такой команды нет, но мы ее можем создать. Для этого следует написать так:

```
\newcommand{\eqdef}{\stackrel{\mathrm{def}}{=}}
```

После того как \TeX прочтет эту строку, он всюду, встречая команду `\eqdef`, будет реагировать точно так же, как если бы он видел текст `\stackrel{\mathrm{def}}{=}`.

Например, формула $x^2 \stackrel{\text{def}}{=} x \cdot x$ теперь получается так:

```
x^2\eqdef x\cdot x
```

Новая команда \TeX ’а, которую мы определили, называется макросом (еще говорят: макроопределение, макрокоманда, макро). Рассмотрим точные правила для создания макросов средствами \LaTeX ’а.

Для создания макросов используется команда (`newcommand`). Эта команда имеет два обязательных аргумента. Первый из них — имя, которое вы придумали для вашего макроса. Имена макросов должны подчиняться тем же правилам, что имена \TeX ’овских команд: либо `backslash` и после него одна не-буква, либо `backslash` и после него — последовательность букв. Второй обязательный аргумент команды (`newcommand`), называемый «замещающим текстом», сообщает \TeX ’у смысл макроса: на этот текст ваш макрос будет замещаться в процессе трансляции (как говорят, макрос будет «разворачиваться»).

При пользовании командой `\newcommand` нельзя в качестве имени макроса выбирать имя уже существующей команды или окружения (если вы попытаете так сделать, \LaTeX выдаст сообщение об ошибке).

Во втором аргументе команды `\newcommand` (иными словами, в «замещающем тексте») вместе с каждой открывающей фигурной скобкой должна присутствовать соответствующая ей закрывающая¹, так что определения наподобие

```
\newcommand{\начатькурсив}{\it}
\newcommand{\кончитькурсив}{}}
```

¹Фигурные скобки, входящие в состав команд `\{` и `\}`, в счет при этом не идут.

являются незаконными и приведут лишь к сообщению об ошибке.

Если команда `\newcommand` дана внутри группы, то смысл определяемой ею новой команды будет забыт \TeX 'ом по выходе из группы. Если новая команда определяется в преамбуле, то, естественно, она будет понятна \TeX 'у на протяжении всего документа.