МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
РОССИЙСКОЙ ФЕДЕРАЦИИ
Рубцовский индустриальный институт (филиал)
ГОУ ВПО "Алтайский государственный технический
университет им. И.И. Ползунова"

М.С. Чебарыков

Основы работы в системе LaTeX

Учебное пособие для студентов
направления "Информатика и вычислительная техника"

Рубцовск 2014

УДК 004.915

Чебарыков М.С. Основы работы в системе LaTeX: Учебное пособие для студентов направления "Информатика и вычислительная техника"/ Рубцовский индустриальный институт. - Рубцовск, 2014. - 49 с.

Учебное пособие посвящено изложению основ работы в системе компьютерной верстки LaTeX. Особое внимание уделено оформлению математических статей и формул. Описаны элементы программирования в системе LaTeX. Изложение материала учитывает уровень подготовки студентов второго курса.

Рассмотрено и одобрено
на заседании НМС РИИ
Протокол №9 от 25.12.14 г.

Рецензент: к.т.н., доцент А.В. Шашок

Содержание

ВВЕДЕНИЕ: TEX, LATEX, MIKTEX

TEX – система компьютерной верстки, разработанная американским профессором информатики Дональдом Кнутом в целях создания компьютерной типографии. В нее входят средства для секционирования документов, для работы с перекрестными ссылками.

В отличие от обыкновенных текстовых процессоров и систем компьютерной верстки, построенных по принципу WYSIWYG – What You See Is What You Get («что видишь, то и получишь»), в TEX'е пользователь лишь задает текст и его структуру, а TEX самостоятельно на основе выбранного пользователем шаблона форматирует документ, заменяя при этом дизайнера и верстальщика.

Документы набираются на собственном языке разметки в виде обычных ASCII-файлов, содержащих информацию о форматировании текста или выводе изображений. Эти файлы (обычно имеющие расширение «.tex») транслируются специальной программой в файлы «.dvi» (device independent – «независимые от устройства»), которые могут быть отображены на экране или напечатаны. DVI-файлы можно специальными программами преобразовать в PostScript, PDF или другой электронный формат.

Ядро TEX'а представляет собой язык низкоуровневой разметки, содержащий команды отступа и смены шрифта. Огромные возможности в TEX'е предоставляют готовые наборы макросов и расширений. Наиболее распространенным расширением стандартного TEX'а (наборы шаблонов, стилей и т. д) является созданный Лесли Лэмпортом LATEX. Пакет позволяет автоматизировать многие задачи набора текста и подготовки статей, включая набор текста на нескольких языках, нумерацию разделов и формул, перекрестные ссылки, размещение иллюстраций и таблиц на странице, создание списка используемой литературы и др.

MiKTeX – открытый дистрибутив TEX для платформы Windows. В настоящее время в состав MiKTeX включены:

- текстовый редактор TeXworks;
- компиляторы TeX, pdfTeX, XeTeX и LuaTeX;
- различные варианты TEX: e-TeX, Omega, NTS;
- программы преобразования dvi-файлов в формат PDF;
- полный набор наиболее часто используемых макропакетов, включая LATEX;
- программа просмотра dvi-файлов – Yap;
- другие инструменты и утилиты.

1 Основные понятия

1.1 Исходный файл

Исходный файл с расширением .tex для системы LaTeX содержит непосредственно текст документа вместе со спецсимволами и командами оформления текста. Этот файл можно создать любым текстовым редактором, включая редактор TeXworks, свободно распространяемый вместе с MiKTeX'ом. Текстовый файл не должен содержать шрифтовых выделений, разбивки на страницы, переносов (TeX сделает их сам) и т. п. Слова отделяются друг от друга пробелами, соседние абзацы – пустыми строками, при этом TeX не различает, сколько именно стоит пробелов или пустых строк.

1.2 Спецсимволы

Большинство символов в исходном тексте прямо обозначает то, что будет напечатано. Следующие 10 символов:

```
{  }$ & # % _ ^ ~ \
```

имеют особый статус. Печатное изображение знаков, соответствующих первым семи из них, можно получить, если в исходном тексте поставить перед соответствующим символом без пробела знак \ (по-английски он называется «backslash»). Ниже приведем пример.

фрагмент исходного текста:	после компиляции в LaTeX'е:
`5 \% , \$ 200.`	5 % , $ 210.

Назначение части из этих спецсимволов указано в таблице ниже.

Символ	Назначение
%	символ комментария – все символы, расположенные в строке после него, TeX игнорирует (в том числе сам %)
{}	ограничивают группы в исходном файле
$	ограничивает математические формулы
_, ^, \, <, >, |	употребляются при записи математических выражений
\	с этого символа начинаются все команды TeX

1.3 Команды и их задание в тексте

Структура команды TeX предельно проста: команда начинается с символа \, после без пробела пишутся имя команды и ее аргументы, если они необходимы. В фигурных скобках в большинстве команд указывается список *обязательных аргументов*, в квадратных скобках – список необязательных аргументов, называемых *опциями*.

С точки зрения записи в исходном тексте, команды принято делить на два типа. Первый тип – команды, начинающиеся со знака \ и одного символа после него, не являющегося буквой. Именно к этому типу относятся команды `\{, \}, \%`. У команд второго типа имя состоит из последовательности букв. Например, команды `\TeX, \LaTeX и \LaTeXe` генерируют эмблемы систем TEX, LATEX и LATEX 2_ε.

Имя команды второго типа нельзя разрывать при переносе на другую строку. В именах команд прописные и строчные буквы различаются. Например, `\large`, `\Large` и `\LARGE` – это три разные команды, они задают различные размеры шрифта.

После команды первого типа (из \ и не-буквы) пробел в исходном тексте ставится или не ставится в зависимости от того, что вы хотите получить на печати. После команды второго типа в исходном тексте обязательно должен стоять пробел либо символ, не являющийся буквой (он указывает компилятору LATEX'а на конец имени команды). В качестве такого символа можно использовать фигурные скобки `{}`.

Если после команды из \ и букв в исходном тексте следуют пробелы, то при трансляции они игнорируются. Чтобы заставить TEX все-таки «увидеть» пробел после команды в исходном тексте (например, чтобы сгенерированное с помощью команды слово не сливалось с последующим текстом), надо этот пробел специально организовать. Один из возможных способов — поставить после команды пару из открывающей и закрывающей фигурных скобок (так что TEX будет знать, что имя команды кончилось), и уже после них сделать пробел, если нужно. Иногда можно также поставить команду \ (backslash с пробелом после него), генерирующую пробел. Вот пример:

фрагмент исходного текста:	после компиляции в LATEX:
`Освоить \LaTeX\ проще, чем \TeX. Человека, который знает систему \TeX{} и любит ее, можно назвать \TeX ником.`	Освоить LATEX проще, чем TEX. Человека, который знает систему TEX и любит ее, можно назвать TEXником.

В последней строке этого примера мы не создали пробела после команды TEX, чтобы эмблема TEX'а слилась с последующим текстом.

Необязательных аргументов в командах может быть несколько; иногда они должны располагаться до обязательных, иногда после. В любом случае порядок, в котором должны идти аргументы команды, надо строго соблюдать. Между скобками, в которые заключены обязательные аргументы, могут быть пробелы, но не должно быть пустых строк.

Некоторые команды имеют *альтернативную форму*, которая используется, если после имени команды указать символ * («звездочка»). Например,

команда перехода на следующий абзац имеет две формы: \\ – переход на новый абзац (с красной строки), * – переход на новую строку без перехода на новый абзац (без красной строки).

1.4 Структура исходного текста

Текст исходного tex-файла можно разделить на две основных части: *преамбулу*, где приводятся команды, относящиеся ко всему документу, и *тело документа*, где находится непосредственно текст документа.

Команды преамбулы указывают класс документа, подключаемые пакеты, глобальные параметры оформления страницы, величины отступов и интервалов и т.д. Преамбула, как и сам LaTeX-файл, должна начинаться с команды `documentclass{класс_документа}`, задающей класс оформления документа.

Существует четыре класса документов (`класс_документа`), которые могут быть присвоены по умолчанию (табл. 1).

Таблица 1

Стандартные классы документов в системе LaTeX

Класс	**Описание**
article	Статья для научных журналов, короткие доклады, письма и пр.
report	Документ, состоящий из нескольких глав (диссертация, небольшая книга)
book	Книга
slide	Слайды для проектора. В этом случае текст выводится большими буквами

Можно создавать и новые классы. Каждый класс определяет свой стиль оформления по умолчанию. Если часть параметров, относящихся ко всему документу (например, величину абзацного отступа), необходимо изменить, то соответствующие команды следует указывать после команды `\documentclass`, до начала тела документа.

Тело документа начинается командой `\begin{document}`. Только после этой команды может идти собственно текст. Заканчиваться файл должен командой `\end{document}`. Любой текст после этой команды компилятор LaTeX'а просто проигнорирует.

Пример простейшего LaTeX-файла, составленный по всем правилам:

```
\documentclass {article}
\begin{document}
Привет, мир!
\end{document}
```

1.5 Группы

Важным понятием TEX'а является понятие *группы* – части текста, заключенной в фигурные скобки.

Как правило, задаваемые командами TEX'а изменения различных параметров действуют в пределах той группы, внутри которой была дана соответствующая команда; по окончании группы все эти изменения отменяются и восстанавливается тот режим, который был до начала группы.

Рассмотрим это на примере команды вывода полужирного шрифта `\bf`:

фрагмент исходного текста:	после компиляции в LATEX'е:
Часть текста набрана {\bf полужирным шрифтом} ; далее текст снова обычный.	Часть текста набрана **полужирным шрифтом**; далее текст снова обычный.

Группы могут быть вложены друг в друга. Фигурные скобки в исходном тексте должны быть сбалансированы: каждой открывающей скобке должна соответствовать закрывающая. Некоторые команды, называемые глобальными, сохраняют свое действие и за пределами той группы, где они были употреблены.

1.6 Окружения

Еще одна важная конструкция LATEX'а – это *окружение* (environment). Окружение — это фрагмент файла, начинающийся с текста:
`\begin{имя_окружения}`,
где имя _ окружения представляет собой первый обязательный (и, возможно, не единственный) аргумент команды `\begin`.

Заканчивается окружение командой `\end{имя_окружения}` (команда `\end` имеет только один аргумент – имя завершаемого ею окружения). Каждой команде `\begin`, открывающей окружение, должна соответствовать закрывающая его команда `\end` (с тем же именем окружения в качестве аргумента).

Важным свойством окружений является то, что они действуют и как фигурные скобки: часть файла, находящаяся внутри окружения, образует группу.

1.7 Параметры

TEX в каждый момент обработки исходного текста учитывает значения различных *параметров*, таких, как величина абзацного отступа, ширина и высота страницы, расстояние по вертикали между соседними абзацами и т.д.

Параметры TEX'а обозначаются аналогично командам: с помощью символа \ («backslash»), за которым следует либо последовательность букв, либо одна не-буква. Чтобы изменить параметр, надо написать его обозначение, а затем, после знака равенства, значение, которое мы «присваиваем» этому параметру: `\имя_параметра=величина`.

Например, `\parindent` обозначает в TEX'е величину абзацного отступа; чтобы абзацный отступ равнялся двум сантиметрам, можно написать так: `\parindent=2cm`.

Это изменение распространяется на текущую группу, если указано в теле одной из групп документа, но может быть применено ко всему документу, если указано в преамбуле.

Для изменения параметров также можно использовать следующие две команды:

```
\setlength{parameter}{length}
\addtolength{parameter}{length}
```

Первая команда присваивает величине parameter значение length, а вторая добавляет к величине parameter значение length. Например, для того чтобы увеличить абзацный отступ на 1 см, необходимо выполнить следующую команду:

`\addtolength{\parindent}{1cm}`.

1.8 Единицы длины

В зависимости от назначения параметра в качестве величины может быть просто число или число с указаниями единицы измерения, часть из которых приведена в таблице ниже:

pt	пункт ≈ 0.35 миллиметра
pc	пика=12pt
mm	миллиметр
cm	сантиметр=10mm
in	дюйм=25,4mm
dd	пункт Дидо ≈ 1,07pt
cc	цицеро=12dd

Можно задавать размеры с помощью любой из этих единиц; при записи дробного числа можно использовать как десятичную запятую, так и десятичную точку; прописные и строчные буквы в обозначениях единиц длины не различаются.

Даже если длина, которую вы указываете TEX'у, равна нулю, все равно необходимо указать при этом нуле какую-нибудь из используемых TEX'ом единиц длины.

Кроме перечисленных, в TEX'е используются еще две «относительные» единицы длины, размер которых зависит от текущего шрифта. Это em, приблизительно равная ширине буквы «M» текущего шрифта, и ex, приблизительно равная высоте буквы «x» текущего шрифта. Эти единицы удобно использовать в командах, которые должны работать единообразно для шрифтов разных размеров.

2 Преамбула документа

2.1 Основные команды преамбулы документа

Теперь подробней остановимся на командах, которые могут стоять в преамбуле документа. Начнем с первой строки входного файла – команды:

`\documentclass[options]{class}`.

Напомним, что эта команда специфицирует, какой документ должен быть в результате получен: аргумент в фигурных скобках class определяет класс документа, а аргумент в квадратных скобках (options) управляет его характеристиками.

Кроме стандартных классов (см. пункт 1.4) могут применяться и классы, задаваемые пользователем, а также классы и стили различных издательств и редакций. Например, для того, чтобы использовать в документе класс статьи Американского математического общества, документ должен начинаться командой:

```
\documentclass{amsart}
```

Параметры (options) команды `\documentclass` приведены в табл. 2

Таблица 2

Параметры стандартных классов документов

Параметр	**Описание**
10pt, 11pt, 12pt	Размер основного шрифта документа (по умолчанию – 10 pt)
a4paper, letterpaper, b5paper	Размер бумаги для печати документа (по умолчанию – letterpaper)
Fleqn	Отображает формулы с выравниванием влево вместо центрирования
leqno	Размещает номера формул слева вместо размещения справа
landscape, portrait	Задает альбомную или портретную ориентацию бумаги (по умолчанию – portrait)
titlepage, notitlepage	Определяет, начинать ли текст после заголовка документа с новой страницы или нет
twocolumn	Вывод текста в две колонки
twoside, oneside	Задает выводную структуру документа для двусторонней или односторонней печати
openright, openany	Разрешает начинать главы только на нечетных страницах или на любых

Базовый набор команд LaTeX не всегда достаточен для решения поставленных задач. Если требуется включение графических объектов в различных форматах, использование различных языков, цветного текста и т. п., то

нужно использовать расширения стандартного набора, которые называются пакетами. Для активации пакета используется команда:
`\usepackage[options]{package}`.
Здесь package является именем пакета, а options – его аргументами, которые, вообще говоря, не обязательны. Некоторые пакеты входят в состав базового сопровождения LaTeX'а, а многие распространяются отдельно. Приведем имена некоторых, наиболее распространенных пакетов в табл. 3, сопровождая их короткими описаниями.

Таблица 3

Пакеты системы LaTeX

Пакет	**Описание**
babel	Пакет языковой поддержки, в том числе и русификации
inputenc	Пакет задания кодировки текста документа (Win, KOI,...)
graphicx	Пакет для импорта в документ рисунков в различных форматах
makeidx	Пакет для создания предметных и именных указателей
amsmath, amsopn, amssymb	Пакеты для оформления математических статей от Американского математического общества
hyperref	Пакет для создания гиперссылок в итоговом документе
maple2e	Пакет для распознавания текста, подготовленного пакетом Maple.

Отметим, что при этом все файлы поддержки пакета (с расширением .sty) должны размещаться или в каталоге самого документа, или в каталоге LaTeX'а со стилевыми файлами. Кроме того, в преамбуле документа могут находиться команды, управляющие параметрами страницы, команды, определяющие правила переноса слов, и др. Обо всем этом будет сказано далее.

2.2 Стили и параметры страницы

Результат обработки текста документа располагается в прямоугольной области на физической странице, которая называется *логической страницей*. Сверху страницы находится верхний колонтитул, а снизу – нижний колонтитул. Отступы сверху и снизу, слева и справа от логической страницы называются *полями* и иногда используются под небольшие фрагменты текста – так называемые *заметки на полях*. В LaTeX'е предусмотрено три стандартных стиля страниц, приведенные в табл. 4. Для их определения используются следующие две команды:
`\pagestyle{style}`
`\thispagestyle{style}`
Первая команда указывает на стиль страницы для всего документа, а вторая позволяет изменять стиль текущей страницы.

Таблица 4

Стандартные стили страницы

Стиль	**Описание**
plain	Стиль, используемый пакетом по умолчанию. Номер страницы печатается по центру нижнего колонтитула
headings	В верхнем колонтитуле печатается заголовок текущей главы и номер страницы
empty	Отсутствие колонтитулов и номеров страниц

Геометрические параметры страницы также могут легко изменяться. Параметры страницы и соответствующие команды вместе со значениями, принятыми по умолчанию для листа размера letter, схематически представлены на рис. 1. Далее даны значения, принятые по умолчанию (они указаны на рисунке под соответствующими номерами):

1. `one inch + \hoffset ,`
2. `one inch + \voffset ,`
3. `\oddsidemargin - 31 pt,`
4. `\topmargine - 28 pt,`
5. `\headheight - 12 pt,`
6. `\headsep - 20 pt,`
7. `\textheight - 526 pt,`
8. `\textwidth - 372 pt,`
9. `\marginparsep - 7 pt,`
10. `\marginparwidth - 71 pt,`
11. `\footskip - 36 pt,`
12. `\marginparpush - 5 pt,`
13. `\hoffset - 0,`
14. `\voffset - 0.`

По умолчанию для нумерации страниц используются арабские цифры. Стиль нумерации можно переопределить с помощью команды

```
\pagenumbering{num_style}
```

Стили `num_style` приведены в табл. 5.

Таблица 5

Стили нумерации

Стиль	**Вид нумерации страницы**
Arabic	Арабские цифры
Roman или roman	Большие или маленькие римские цифры
Alph или alph	Большие или маленькие буквы

Напомним, что размер бумаги указывается с помощью опций команды `\documentclass`.

Иногда возникает необходимость указать другие параметры страницы для печати (см. пункт 1.7). Напомним, что команды, изменяющие параметры страницы, должны находиться в преамбуле документа.

Кроме описанных команд в преамбуле документа могут размещаться команды, определяемые пользователем (см. раздел 6), и некоторые другие команды.

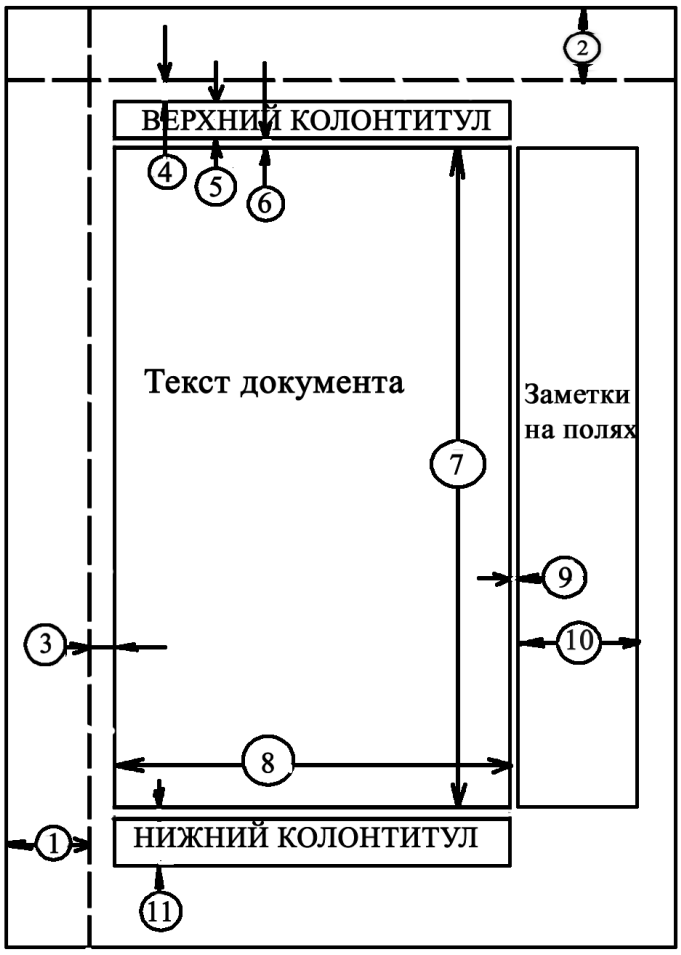

Рис. 1. Параметры страницы

Приведем пример текста документа на русском языке в кодировке Windows-1251 в системе MiKTeX:

```
\documentclass[12pt,a4paper]{article}
\usepackage[cp1251]{inputenc}
\usepackage[russian]{babel}
\addtolength{\textwidth){1cm}
\pagestyle{empty}

\begin{document}
```
Текст данной статьи.
```
\end{document}
```

3 Простейшее оформление тела документа

3.1 Заголовок документа

Документ обычно начинается с заголовка, который определяет группа команд, приведенных в табл. 6.

Таблица 6

Стандартные команды заголовка документа

Команда	**Описание**
`\title`	Определяет название документа
`\author`	Определяет имена авторов документа
`\email`	Определяет электронную почту авторов
`\date`	Позволяет задать дату написания документа; если команда не используется, то LaTeX использует системную дату
`\thanks`	Создает сноску с благодарностями внизу страницы

Эти команды будут активированы, если после них стоит команда создания заголовка: `\maketitle`. Если стиль документа *article*, то заголовок документа размещается перед текстом на первой странице, а для остальных стилей создается отдельный титульный лист.

3.2 Разделы, главы, абзацы, примечания

В системе LaTeX существует возможность разделять текст на разделы с различным уровнем вложенности при помощи команд, приведенных в табл. 7 (сверху вниз – от самого высокого уровня вложенности до самого низкого).

Таблица 7

Команды создания разделов документа

Команда	**Описание**
`\part [opt]{text}`	Часть.Употребляется только для класса book.
`\chapter [opt]{text}`	Глава. Употребляется только для класса book.
`\section [opt]{text}`	Раздел.
`\subsection [opt]{text}`	Подраздел.
`\subsubsection [opt]{text}`	«Подподраздел».
`\paragraph [opt]{text}`	Текст пользователя выделяется отступами и шрифтом.
`\subparagraph [opt]{text}`	Самый низкий уровень вложенности.
`\appendix`	Эта команда не выводит никакого текста, а служит для обнуления счетчиков.

Во всех перечисленных командах, кроме последней, аргумент *text* содержит заголовок соответствующего раздела. Эти команды используются для автоматической нумерации разделов, а также создания оглавления документа и колонтитулов. Опция opt определяет альтернативный заголовок (обычно более короткий) для замены текста заголовка *text* в оглавлении и колонтитулах.

Команда `\appendix` полезна для создания приложений в конце документа, поскольку они обозначаются иным способом, чем разделы основного документа.

Для определения сносок внизу страницы существует команда `\footnote[number]{text}`, где *text* – текст сноски, а *number* является необязательным параметром, предназначенным для изменения номера сноски.

Примечания на полях можно сделать при помощи команды создания заметок – `\marginpar{text}`. По умолчанию текст выводится в правом поле страницы. Для того чтобы сменить поле для заметок на левое, используется команда `\reversemarginpar`, а для возвращения значения по умолчанию надо выполнить команду `\normalmarginpar`.

Команда перехода на новый абзац – `\par` (сокращенная форма – `\\`), переход на новую строку без изменения абзаца – `\\*`. Команду `\\` можно заменить одной или несколькими пустыми строками. Отметим, что пустая строка не должна встречаться в математических формулах, это приведет к ошибке при компиляции.

По умолчанию пакет выравнивает абзацы по ширине. Окружения flushleft, center, flushright служат для выравнивания абзаца по левому краю, по центру и по правому краю соответственно.

По умолчанию первая строка для первого в разделе абзаца печатается без отступа (так принято в США), а первые строки для последующих абзацев – с отступом. Однако эти правила можно изменить при помощи команд управления абзацным отступом: `\indent` – производит горизонтальный отступ в начале абзаца, `\noindent` – подавляет горизонтальный отступ в начале абзаца. Можно также подключить пакет `indentfirst` в преамбуле, чтобы первая строка для первого в разделе абзаца начиналась с красной строки, как принято в России и других европейских странах.

3.3 Разрывы, интервалы, переносы

При создании печатного варианта LaTeX автоматически разбивает документ на страницы, сам выравнивает строки и делает переносы. Но часто возникают ситуации, когда необходимо это сделать принудительно, если автоматический вариант не удовлетворяет нашим пожеланиям. Для управления разрывами существуют команды, приведенные в табл. 8.

Опция number принимает значения между 0 и 4 и управляет степенью обязательности применения команды: значение 4 соответствует наибольшей обязательности, а 0 позволяет LaTeX'у игнорировать команду, если ее применение приводит к плохому результату.

Команда `\enlargethispage{size}` позволяет изменить размер поля текста одной из страниц, на небольшую величину аргумента size.

Несколько пустых строк LaTeX считает за одну, несколько пустых пробелов – за один пробел. Расстояние между словами в строке или абзацами в документе пакет определяет автоматически, а для принудительного изме-

нения интервала используются команды `\hspace{length}` – горизонтальный отступ длиною length, `\vspace{length}` (сокращенная форма `\\[length]`) – вертикальный интервал длиною length. В последних трех командах length является длиной отступа в любых единицах, разрешенных в системе LaTeX (см. пункт 1.8).

Таблица 8

Команды управления разрывами абзацев и строк

Команда	**Описание**
`\\`или `\newline`	Конец строки и переход на новый абзац
`\\*`	Конец строки без перехода на новый абзац
`\newpage`	Переход на новую страницу
`\pagebreak [number]`	Разрыв страницы
`\nopagebreak [number]`	Запрет перехода на новую страницу
`\linebreak [number]`	Переход на новую строку
`\nolinebreak [number]`	Запрет перехода на новую строку

Для запрета автоматического изменения интервала между словами служит символ «~». Эту команду можно использовать также в качестве неразрывного пробела. Например, инициалы не будут отрываться от фамилии, если написать так: Иванов~И.~И.

Следующие три команды должны находиться в преамбуле документа. Команда задания межстрочного интервала – `\linespread{size}`. Здесь *size* определяет размер интервала: *size*=1.3 соответствует полуторному интервалу, *size*=1.6 – двойному, по умолчанию задан одинарный интервал (*size*=1).

Команда `\setlength{\parindent){value}` – задает расстояние величины value отступа в первой строке абзаца, а `\setlength{\parsk1p}{value}` – определяет интервал между абзацами.

Расстановкой переносов и плотностью строки управляют команды `\fussy` и `\sloppy`. Первая команда активна по умолчанию и приводит к тому, что транслятор старается оставить как можно меньше лишних пробелов в строке. Команда `\raggedright`, напротив, позволяет установить режим без выравнивания и без переносов.

LaTeX поддерживает алгоритм автоматической расстановки переносов на основе встроенных словарей. Пользователь может дополнить базовые правила переносов при помощи команды `\hyphenation{список_слов}`.

Эта команда должна находиться в преамбуле документа. В скобках даются слова, для которых определяются правила переносов, причем места возможных переносов помечаются символом «-», а слова отделяются друг от друга пробелами. Например:

`\hyphenation{na-pa-маг-не-тик син-гу-ляр-ность}`.

Существует еще возможность жесткого принудительного переноса командой \-. Пакет всегда будет делать перенос на месте, помеченном эти-

ми символами, конечно, если слово стоит в конце строки. Если же слово или фразу нельзя разрывать, то для запрещения переноса служит команда `\mbox{text}`.

3.4 Шрифты, размеры, специальные и национальные символы

LaTeX автоматически выбирает подходящие шрифты, исходя из стиля документа. Но начертание и размер шрифта можно установить вручную. Для изменения шрифта текста, стоящего в фигурных скобках, имеются команды, приведенные в табл. 9.

Таблица 9

Команды изменения шрифта текста

Команда	**Название шрифта**
`\textrm {...}`	roman
`\textsf {...}`	sans serif
`\texttt {...}`	`typewriter`
`\textmd {...}`	medium
`\textbf {...}`	**bold face**
`\textup {...}`	upright
`\textit {...}`	*italic*
`\textsl {...}`	*slanted*
`\textsc {...}`	SMALL CAPS
`\emph {...}`	*emphasised*
`\textnormal {...}`	document font

Изменение размера символа шрифта в системе LaTeX имеет определенные особенности. Реальный размер символов будет зависеть от базового размера шрифта, указанного явно или по умолчанию для данного класса документа. Все остальные шрифты изменяются пропорционально ему. Размеры символов изменяются после применения команд, которые даны в табл. 10.

Иногда требуется выделить отдельные фразы на фоне остального текста. Для выделения текста курсивом на фоне прямого шрифта и прямым шрифтом на фоне курсива служит команда `\emph{text}`. Команды подчеркивания (`\underllne{text}`) и заключение части текста в рамку (`\fbox{text}`) также могут использоваться для выделения отрывка текста.

Существует возможность воспроизводить тире различной длины. Это осуществляется через повторение символа «-» (минус), причем длина тире зависит от количества минусов. Для набора многоточия используется команда `\ldots`. Знак параграфа набирается с помощью команды «`\S`», а знак © – командой `\copyright`. Наконец, любой символ можно набрать, зная его код, при помощи команды `symbol{код}`.

Таблица 10

Команды управления размером символа

Команда	**Результат**
`\tiny`	<tiny>самый маленький шрифт</tiny>
`\scriptsize`	<scriptsize>очень маленький шрифт</scriptsize>
`\footnotesize`	<footnotesize>шрифт для сносок</footnotesize>
`\small`	маленький шрифт
`\normalsize`	базовый шрифт
`\large`	<large>шрифт чуть больше базового</large>
`\Large`	<Large>еще больше</Large>
`\LARGE`	<LARGE>очень большой шрифт</LARGE>
`\huge`	<huge>огромный</huge>
`\Huge`	<Huge>самый большой</Huge>

Список в табл. 11 показывает, как получать буквы и символы национальных алфавитов. В каждой из четырех колонок символ дан слева, а справа – соответствующая команда.

Таблица 11

Символы национальных алфавитов

ō – `\=o`	ȯ – `\.o`	ö – `\o"`	ç – `\c c`
ŏ – `\u o`	ǒ – `\v o`	ő – `\H o`	ǫ – `\c o`
ó – `\'o`	ô – `\^o`	ọ – `\d o`	o̲ – `\b o`
o͡o – `\t oo`	œ – `\oe`	Œ – `\OE`	æ – `\ae`
Æ – `\AE`	å – `\aa`	Å – `\AA`	!’ – `!’`
ø – `\o`	Ø – `\O`	ł – `\l`	Ł – `\L`

Несколько слов о поддержке языков, отличных от английского, в частности о русификации. Вообще говоря, подключение языковой поддержки зависит от конкретной реализации пакета. Наиболее распространенным в настоящее время является пакет *babel*, который должен подключаться при помощи команды `\usepackage[language]{babel}`. Кроме того, для некоторых языков, в частности для русского, существует несколько вариантов кодировок символов (KOI-8, Windows,...). В этом случае кодировка должна быть задана при помощи подключения пакета *inputenc*.

3.5 Колонки, цитаты, блоки

По умолчанию документ будет печататься в одну колонку. Если необходимо печатать в две колонки весь документ, то надо указать параметр twocolumn в команде `\documentclass`. Если же выводить в две колонки надо

только часть текста, то используется команда `\twocolumn` внутри документа. После этой команды с новой страницы текст будет выводиться в две колонки. Кроме того, существуют и другие возможности вывода текста в несколько колонок (таблицы, блоки).

Для того чтобы набрать часть текста с отступом от краев (важная фраза, цитата и др.), надо использовать окружение quote. Существуют еще два похожих окружения – quotation и verse.

Любой текст внутри окружения `verbatim` напечатается «текстом пишущей машинки», т.е. так же, как и в исходном файле, и никакие команды LaTeX'а внутри окружения не будут восприниматься транслятором.

Например, рассмотрим следующий исходный текст:
`\begin{verbatim}` `Пример команды \LaTeX'a: \ldots` `\end{verbatim}`
После обработки LaTeX'ом текст останется неизменным:

```
Пример команды \LaTeX'a: \ldots
```

То же предложение вне окружения `verbatim` будет воспринято транслятором иначе (выполнятся команды `\LaTeX` и `\ldots`):
Пример команды LaTeX'a: ...

Основным объектом, с которым работает транслятор пакета, является блок, в качестве которого может фигурировать как одна буква, так и целые фразы. Выше была рассмотрена команда `\mbox`, которая делает неразрывной целую фразу, точнее, делает ее как бы одним словом.

Существует команда, в которой можно указывать ширину блока:
`\makebox[length][par]{text}`.
Аргумент length указывает на длину строки в блоке, причем длина может задаваться как единицами размера, так и при помощи слова или последовательности символов, text – текст, находящийся в блоке. В качестве опции par могут выступать символы r, l, c, означающие соответственно выравнивание влево, вправо и по центру.

Для создания блока переменной высоты используется команда:
`\parbox[pos]{length}{text}`.
Необязательный аргумент pos определяет вертикальное позиционирование блока и может принимать следующие значения: t – верхняя строка блока выравнивается по текущей строке, c – середина блока выравнивается по середине текущей строки, b – нижняя строка блока выравнивается по текущей строке.

Для заключения в рамку фрагмента текста применяется команда `\fbox`. Существует также команда `\framebox` для взятия текста в рамку, назначение параметров которой совпадает с описанным для команды `\makebox`.

Описанные блочные команды используются обычно для небольших фрагментов текста. Для создания больших блоков, включающих различные объекты (таблицы или рисунки), существует окружение `minipage`, параметры

которого аналогичны параметрам окружения `parbox`:
`\begin{minipage}[pos]{width}`
...
`\end{minipage}`.

4 Оформление различных элементов текста

4.1 Списки

Структура простого списка в LaTeX'е выглядит так:

```
\begin{имя_окружения}
\item[опции] элемент 1,
\item[опции] элемент 2,
...
\end{имя_окружения}
```

Для создания маркированных списков применяется окружение itemize, для нумерованных списков следует прибегнуть к окружению enumerate. Если же создается перечень, где каждый элемент имеет свой заголовок, то используется окружение description. Во всех трех случаях каждый элемент тела списка начинается командой `\item`, которая может иметь необязательные аргументы. Например, для окружения description в качестве опций будут выступать заголовки соответствующих элементов.

Отметим, что до первой команды в перечнях могут стоять только команды управления шрифтами и подобные им, но не должно быть никакого текста. Перечни могут быть вложенными друг в друга. Максимальная глубина вложенности равна четырем. Примеры перечней приведены ниже:

Маркированный список	**Нумерованный список**	**Список-заголовок**
• элемент 1,	1. элемент 1,	**заголовок 1** элемент 1,
• элемент 2.	2. элемент 2.	**заголовок 2** элемент 2.

4.2 Ссылки и нумерация

LaTeX позволяет организовать перекрестные ссылки на страницы документа или его разделы так, чтобы компилятор автоматически определял номер страницы для ссылки. Для этого в месте, на которое нужно ссылаться, ставится метка при помощи команды: `\label{name}`, где name служит именем, на которое в файле можно ссылаться одним из следующих способов:
`\pageref{name}` или `\ref{name}`

Первая из этих команд ссылается на номер страницы, на которой стоит метка, а вторая – на номер главы, секции, формулы или другого нумерованного объекта.

Если в файле есть ссылки, то для правильной его обработки компилятор необходимо запускать два раза. После первого прохода LaTeX записывает информацию о ссылках в специальный файл, а в выходном файле на месте ссылок могут появиться знаки вопроса – ??. При повторной компиляции информация о ссылках считывается, и в выходном файле все ссылки будут отображены правильно. Отметим также, что некоторые тестовые редакторы для работы в LaTeX'е (в том числе и поставляемый с MiKTeX'ом редактор TeXworks) автоматически дважды или даже трижды компилируют исходный файл при выборе соответствующего пункта меню.

Если подключить в преамбуле пакет hyperref, то команды `\pageref` и `\ref` будут переопределены и все перекрестные ссылки LaTeX'а автоматически превратятся в гипертекстовые. В этом случае гипертекстовыми ссылками становятся все элементы оглавления, списков таблиц и рисунков, а также маркеры сносок и элементов цитированной литературы. Кроме того, все номера страниц в предметном указателе также становятся гипертекстовыми ссылками.

Для гиперссылок в pdf-файле, когда не требуется вручную листать документ, можно обойтись без номера раздела. Пакет nameref (загружается пакетом hyperref) вводит команду `\nameref{name}`, которая делает текстом гиперссылки на метку name название раздела, в котором находится метка.

Еще один подключаемый пакет `xr-hyper` позволяет создавать гиперссылки на элементы другого документа. Его следует загружать перед пакетом hyperref. После этого описанные выше команды создают гиперссылки на метки, созданные в других документах командой `\label`. Имена aux-файлов этих документов необходимо объявить в преамбуле текущего входного файла посредством деклараций
`\externaldocument[prefix-]{file}[URL]`.
Здесь file — имя файла без расширения. Опция prefix позволяет исключить совпадение имен меток в разных документах. При наличии опции все ссылки в текущем документе на объекты из внешних файлов, помеченные как `\label{name}`, записываются в виде `\ref{prefix-name}`. Опция URL указывает адрес готового документа.

Команда `\href{URL}{text}` позволяет создавать гиперссылку на документ любого типа, хранящийся в сети по адресу URL. Аргумент text оформляется как текст гиперссылки.

Многие объекты LaTeX нумерует автоматически. Например, каждой главе, секции или отдельной формуле соответствует номер. Эта операция осуществляется при помощи *счетчиков*, как стандартных, так создаваемых пользователем (подробнее см. раздел 6). Имя счетчика обычно совпадает с именем использующего его окружения или команды.

Таблица 12

Имена стандартных счетчиков

part	paragraph	figure
chapter	subparagraph	table
section	page	equation
subsection	mpfootnote	
subsubsection	footnote	

Чтобы вывести номер счетчика, используются команды вида `\thename`. Здесь name является именем счетчика. Например, чтобы вывести номер страницы, нужно выполнить команду `\thepage`. Значение счетчика обычно является целым числом. В зависимости от команды задаваемое счетчиком число может выводиться арабскими, римскими цифрами или специальными символами. Команды, работающие со счетчиками, сначала изменяют их значение, а потом используют. Большинство счетчиков до начала работы равны нулю.

Большинство счетчиков автоматически увеличивают свое значение на единицу, когда компилятор встречает в тексте исходного файла одноименную команду, связанную со счетчиком. Счетчики могут быть иерархически связаны друг с другом (например, счетчики разделов документа), в этом случае в момент изменения высшего по иерархии счетчика все подчиненные ему счетчики обнуляются. Так, в начале нового раздела (section) обнуляются все счетчики подразделов (subsection, subsubsection).

Для непосредственного изменения значения счетчика существуют две команды: `\setcounter{name}{num}` – устанавливает счетчику с именем name значение num, `\addtocounter{name}{num}` – увеличивает значение счетчика name на величину num.

4.3 Плавающие объекты: рисунки и таблицы

Оптимальным образом размещать рисунки и таблицы в документе помогает введение плавающих объектов, которые должны полностью находиться на одной странице. Плавающий объект автоматически размещается пакетом в том месте, где он помещается целиком. Существуют два стандартных окружения, позволяющие формировать такие объекты:

`\begin{figure}[loc]`...`\end{figure}`

`\begin{table)[loc]` ...`\end{table}`

Здесь необязательный аргумент *loc* определяет способ размещения плавающего объекта и может состоять из последовательности следующих символов: *h* означает, что объект надо разместить после текущей строки, *p* — разместить на отдельной странице, *t* — наверху страницы, *b* — внизу страницы. По умолчанию пакетом устанавливается последовательность *tbp*. Кроме того, существует параметр *(!)*, который ослабляет ограничения, если стоит вместе с одним из вышеописанных аргументов.

Для нанесения подписей к плавающим объектам используется команда `\caption[entry]{head}`. Здесь необязательный аргумент *entry* используется для составления списка рисунков или таблиц, а аргумент *head* определяет текст подписи. Этот текст не должен быть разделен на абзацы и не может превышать двухсот символов.

Иногда плавающие объекты автоматически располагаются на нежелательной странице. В этом случае такое их размещение можно запретить при помощи команды `\suppressfloats[noloc]` Опция *noloc* может принимать значения *t* или *b*.

4.3.1 Рисование в LaTeX'е

В систему LaTeX включен набор команд, которые позволяют создавать несложные рисунки или чертежи. Для этого существует следующее окружение:

```
\begin{picture}(width, height)(x0,y0)
...
\end{picture}
```

Первая пара аргументов этого окружения обязательна и определяет соответственно высоту и ширину рисунка. Единица измерения LaTeX задается переменной *unitlength*, которая по умолчанию имеет значение *lpt*. Вторая (необязательная) пара аргументов задает координаты левого нижнего угла рисунка. При отсутствии этих параметров считается, что левый нижний угол имеет координаты (0,0).

Внутри окружения можно использовать специальные графические команды, некоторые из которых перечислены в табл. 13.

Графика для LaTeX'а создавалась давно и, естественно, уже устарела, однако существует возможность включения в документ рисунков, созданных другими, более развитыми средствами.

4.3.2 Включение графических файлов

LaTeX позволяет вставлять рисунки, находящиеся в графических файлах наиболее распространенных форматов (jpg, png, eps, ps и др.), причем набор допустимых форматов зависит от операционной системы и программы-компилятора. Поэтому желательно использовать аппаратно-независимые форматы, такие как *PostScript*. Существует целый ряд расширений LaTeX'а для включения графических файлов, но мы остановимся на распространенном пакете графики *graphicx*.

Вставка рисунка из графического файла в пакете *graphicx* осуществляется командой `\includegraphics[optionlist]{filename}`. Обязательный аргумент filename определяет имя графического файла с расширением.

Таблица 13

Некоторые графические команды пакета LaTeX

Команда	**Описание**
`\put (x,y){object}`	Определяет точку привязки с координатами (x,y) графического объекта *object* на рисунке. Объектом может быть текст как в блоках, так и без них или графические команды.
`\line (a,b){length}`	Проводит отрезок заданной длины *length* из текущей точки рисунка. Первый параметр, состоящий из пары целых чисел из интервала от -4 до 4, задает наклон линии. Отношение этих чисел определяет тангенс угла наклона отрезка.
`\linethickness {val}`	Определение толщины вертикальных и горизонтальных линий в любых единицах измерения.
`\vector (xs,ys){dx}`	Рисование стрелок. Назначение аргументов аналогично команде line
`\qbezier [num](xa,ya) (xb,yb)(xc,yc)`	Рисование кривой Безье, проходящей через три точки. Необязательный аргумент num задает число выводимых точек кривой, в случае его отсутствия выводится непрерывная кривая. Координаты точек задаются в круглых скобках
`\circle {diam}`	Рисование окружности радиуса *diam* с центром в текущей точке

Аргумент optionlist представляет собой список параметров, которые управляют характеристиками изображения. Эти параметры представляют собой выражения типа равенств, в левой части которых стоит параметр, а в правой – его значение. В табл. 14 перечислены основные допустимые параметры.

Напомним, что обычно рисунки помещают внутри окружения `figure`, что позволяет легко делать подписи к рисункам и автоматически их нумеровать.

Приведем пример с перечисленными командами.

```
% импортирование векторной графики
\includegraphics{test.eps)
\includegraphics[angle=15,height=3cm]{test.eps}
% импортирование растровой графики
\includegraphics[bb = 0 0 3.5cm 3cm]{test.jpg}
```

В первом случае импортируется векторная графика PostScript-файла (файл test.eps), из которого LaTeX сам берет информацию о размере изображения. Отметим, что при вводе растровой графики, например, jpg-файлов, необходимо указывать размеры рисунка (см. последняя команда примера).

Таблица 14
Аргументы команды импортирования графических файлов

Параметр	**Описание**
bb	Определяет размеры рисунка: четыре числа, которые разделены пробелами. Первые два числа определяют координаты нижнего левого угла, а два других задают координаты правого верхнего угла
natwidth, natheight	Определяют ширину и высоту рисунка в заданных единицах измерения
draft	Устанавливает черновой режим импортирования, когда вместо рисунка выводится рамка по размерам рисунка, а внутри рамки дается имя файла вместо самого рисунка
scale	Изменение масштаба рисунка, например если указано scale=3, то рисунок будет увеличен в 3 раза
width, height	Ширина и высота рисунка при печати
origin	Определяет положение центра вращения рисунка. Вертикальная и горизонтальная координаты центра задаются дискретно и могут принимать следующие значения: горизонтальная координата — l (слева), c (в центре), r (справа); вертикальная координата — t (сверху), c (в центре), b (внизу). Например, *origin=lt* задает вращение вокруг левого верхнего угла
angle	Поворот изображения на заданный параметром *angle* угол вокруг точки, заданной параметром *origin* (по умолчанию вокруг точки привязки). Углы задаются в градусах
keepaspectratio	Если этот параметр имеет значение *true*, то при изменении размеров рисунка сохраняются пропорции

4.3.3 Верстка таблиц

Окружение tabular применяется для создания таблиц, содержащих текст, причем как с рамками, так и без них. Окружение имеет вид:
`\begin{tabular}{spec}` . . . `\end{tabular}`.
В качестве аргумента spec применяются следующие обозначения: l - для колонки с выравниванием влево, r - для колонки с выравниванием вправо, c - для центрированного текста, `p{width}` - для колонки, содержащей текст с разрывами линий, символ «|» используется для проведения вертикальных линий, а команда `\hline` - для горизонтальных.

Переход к новой строке таблицы осуществляется командой \\, а для разделения колонок применяется амперсанд «&». Пробелы в начале и конце графы таблицы игнорируются. Для вставки текста text между столбцами существует команда `@{text}`. Для создания надписей, которые охватывают

несколько колонок, используется команда `\multicolumn`. У этой команды три обязательных параметра: количество охватываемых колонок, тип выравнивания текста в этой графе и сам текст.

Иногда в таблице надо провести короткую горизонтальную линию. Это можно сделать командой `\cline`, имеющей обязательный параметр – номера первой и последней подчеркиваемых колонок. Команда `\vline` проводит вертикальную линию на всю высоту строки.

Приведем два примера. Исходный текст создания простой таблицы:

```
\begin{tabular}{|l|c|r|}
\hline
Глава & Пакет & Назначение \\ \hline
1 & Maple & Символьные вычисления\\
2 & Matlab & Численный анализ\\
3 & \LaTeX & Верстка текстов\\ \hline
\end{tabular}
```

Результат:

Глава	Пакет	Назначение
1	Maple	Символьные вычисления
2	Matlab	Численный анализ
3	LaTeX	Верстка текстов

Пример более сложной таблицы:

```
\begin{tabular}{| l | r@{ m} | r@{ cm}|}
\hline \multicolumn{3}{|c|}{Английские меры длины}\\
\hline & \multicolumn{2}{|c|}{Соответствие} \\ \cline{2-3}
Единицы & \hspace{1.5cm} & \\ \hline
Фут & 0.3 & 30.5 \\ Ярд & 0.9 & 91.44 \\
Миля & 1609 & 160900\\
\hline
\end{tabular}
```

Результат:

Английские меры длины		
	Соответствие	
Единицы	m	cm
Фут	0.3 m	30.5 cm
Ярд	0.9 m	91.44 cm
Миля	1609 m	160900 cm

Для автоматической нумерации и удобства создания подписей таблицы обычно размещаются внутри окружения `table`.

4.4 Оглавления

Под оглавлениями здесь понимаются оглавление документа, а также перечни рисунков и таблиц, которые LaTeX позволяет автоматически генерировать. В процессе работы пакет автоматически собирает информацию о разбивке документа на разделы (секции) и размещении плавающих объектов. Эта информация записывается в служебные файлы в момент обработки команды `\end{document}`. Поэтому для того, чтобы оглавления были сгенерированы правильно, следует входной файл обработать транслятором по крайней мере три раза. Считывают из файлов и печатают оглавление, списки таблиц и рисунков соответственно команды:
`\tableofcontents`,
`\listoffigures`,
`\listoftables`.

Информацию об оглавлении формируют команды секционирования (разбивка документа на разделы, главы и пр.). Информация о рисунках и таблицах задается командой `\caption` внутри соответствующих окружений. В случае нестандартных ситуаций можно записать в оглавлении дополнительные пункты при помощи команды `\addcontentsline{ext}{unit}{entry}`.

Здесь параметр ext определяет, в какое оглавление надо добавить пункт: toc – оглавление документа, lof – список рисунков, lot – список таблиц. В зависимости от значения ext параметр unit может принимать следующие значения: toc – part, chapter и т.д., lof – figure, lot – table. Последний параметр entry задает сам текст строки в оглавлении и, в свою очередь, может содержать некоторые команды.

Команда contentsname определяет заголовок оглавления. Существует еще одна команда, которая добавляет текст в оглавление:
`\addtocontents{ext}{text}`. Первый параметр аналогичен рассмотренному для предыдущей команды, а вторым параметром является добавляемый текст.

4.5 Библиография

Библиография - важный компонент любой статьи или книги. LaTeX позволяет легко готовить списки литературы, состоящие из разнообразных источников и автоматически расставлять ссылки на них в тексте документа. Для печати списка цитируемой литературы в LaTeX'е существует окружение
`\begin{thebibliography}{par}`
`\end{thebibliography}`.

Аргумент par определяет сдвиг левой границы печати списка литературы на ширину текста, стоящего в качестве этого параметра. Номера источников ставятся на месте образовавшегося пробела, потому задаваемая ширина отступа не должна быть меньше, чем ширина печати максимального

номера. Внутри окружения источники перечисляются при помощи команды `\bibitem[lab]{key}`. После этого следует непосредственно текст библиографического источника. Соответствующая запись будет помечена символами lab, а ссылаться на источник можно будет по метке key. При отсутствии метки lab вместо нее будет печататься порядковый номер источника, который хранится в счетчике enumiv.

Для ссылок на источники в тексте документа применяется следующая команда: `\cite[text]{key}`. Результатом этой команды будет печать символов lab или номера источника с меткой key, заключенных в квадратные скобки. Если одной командой нужно сослаться на разные источники, то соответствующие метки перечисляются через запятую. Необязательный параметр text служит для уточнения ссылки текстовым комментарием. Приведем пример документа, содержащего список литературы:

```
    \documentclass[12pt,russian]{article}
\usepackage[cp1251]{inputenc}
\usepackage[russian]{babel}
\textwidth=12cm
\begin{document}
Первой книгой по {\TeX'y} была книга \cite{Knut}, а по {\LaTeX'y} - книга
\cite{Lamport}. Сейчас существует много литературы как о {\TeX'e}, так
и о {\LaTeX'e} (см. \cite{Knut,Lamport}).
\begin{thebibliography}{99999}
\bibitem{Knut} D. E. Knuth. The TeX book Volume A of Computers and
Typesetting. Addison-Wesley Publishing Company. 1964.
\bibitem[\LaTeX]{Lamport} L. Lamport. LaTeX: A Document Preparation
System. Addison-Wesley. Reading. Massachusetts, second edition. 1994.
\end{thebibliography}
\end{document}
```

Результат:

Первой книгой по TEX'у была книга [1], а по LATEX'у – книга [LATEX]. Сейчас существует много литературы как о TEX'е, так и о LATEX'е (см. [1, LATEX]).

Список литературы

1. D. E. Knuth. The TeX book Volume A of Computers and Typesetting. Addison-Wesley Publishing Company. 1964.

LATEX. L. Lamport. LaTeX: A Document Preparation System. Addison-Wesley. Reading. Massachusetts, second edition. 1994.

5 Набор математических выражений в LaTeX'е

В этом разделе рассмотрим одно из главных назначений системы TeX – набор формул. Далее большее внимание будет уделено возможностях базового комплекта пакета LaTeX 2_ε, также будет затронут пакет AMS, который был создан под эгидой Американского математического общества. Для набора математических выражений пакет имеет специальные режимы, которые реализованы в виде окружений и команд.

5.1 Оформление формул

Различают математические формулы внутри текста и так называемые «выключные», то есть выделенные в отдельную строку. Формулы внутри текста задаются окружением math или, что приводит к тем же результатам, обособляются с обеих сторон знаком доллара $.

Выключную формулу можно оформить как окружение displaymath или обособить с обеих сторон парами знаков доллара $$, как предусмотрено стандартом TeX'а. Кроме того, выключную формулу можно задать LaTeX'овскими знаками, введенными Лесли Лэмпортом: `\[` (в начале) и `\]` (в конце). Эти альтернативные обозначения полностью эквивалентны стандартным (со знаками доллара), за одним важным исключением: если выключные формулы обозначаются LaTeX'овскими, а не TeX'овскими обозначениями, то можно сделать так, что выключные формулы будут не центрированы, а прижаты влево.

Пример формулы внутри текста и выключной формулы.

Формула $\int\limits_a^b f(x)\,dx$ находится внутри текста, а следующая формула является выключной:

$$\frac{a+b}{x-y} = g(x).$$

Проставить нумерацию формул можно вручную с помощью команды `\eqno (номер_формулы)`. Пример:`x^2=y \eqno (3.2)`. Результат:

$$x^2 = y. \eqno (3.2)$$

Для автоматической нумерации выключных формул следует использовать окружение equation.

5.2 Таблицы спецзнаков в формулах

В этом разделе мы перечислим математические знаки, используемые LaTeX'ом в формулах. Знаков этих очень много, поэтому разобьем их на несколько групп.

5.2.1 Операции, отношения и просто значки

Начнем с греческих букв. Имя команды, задающей строчную греческую букву, совпадает с английским названием этой буквы (например, буква α задается командой `\alpha`). Исключение составляет буква о (она называется «омикрон»): по начертанию она совпадает с курсивной латинской о, так что специальной команды для нее не предусмотрено, и для ее набора достаточно просто написать о в формуле. Некоторые греческие буквы имеют по два варианта начертаний, например π (`\pi`) и ϖ (`\varpi`), ρ (`\rho`) и ϱ (`\varrho`).

Имя команды, задающей прописную греческую букву, пишется с прописной буквы (например, буква Ψ задается командой `\Psi`). Некоторые прописные греческие буквы (альфа, например) совпадают по начертанию с латинсими, и для них специальных команд нет.

Следующая серия символов – символы, рассматриваемые TEX'ом как символы бинарных операций (на подобие знаков сложения, умножения и т.п.). TEX оставляет в формуле большие пробелы по обе стороны этих знаков, кроме случаев, когда есть основания считать, что эти знаки используются не для обозначения операций, а для других целей.

$+$	`+`	$-$	`-`	$*$	`*`
$\pm$	`\pm`	$\mp$	`\mp`	$\times$	`\times`
$\div$	`\div`	$\setminus$	`\setminus`	$\cdot$	`\cdot`
$\circ$	`\circ`	$\bullet$	`\bullet`	$\cap$	`\cap`
$\cup$	`\cup`	$\uplus$	`\uplus`	$\sqcap$	`\sqcap`
$\sqcup$	`\sqcup`	$\vee$	`\vee`	$\wedge$	`\wedge`
$\oplus$	`\oplus`	$\ominus$	`\ominus`	$\otimes$	`\otimes`
$\odot$	`\odot`	$\oslash$	`\oslash`	$\triangleleft$	`\triangleleft`
$\triangleright$	`\triangleright`	$\amalg$	`\amalg`	$\diamond$	`\diamond`
$\wr$	`\wr`	$\star$	`\star`	$\dagger$	`\dagger`
$\ddagger$	`\ddagger`	$\bigtriangleup$	`\bigtriangleup`	$\bigcirc$	`\bigcirc`
$\bigtriangledown$	`\bigtriangledown`				

В следующей таблице собраны символы "бинарных отношений". Вокруг них TEX также оставляет дополнительные пробелы, но не такие, как вокруг символов бинарных операций.

$<$	`<`	$>$	`>`	$=$	`=`
$:$	`:`	$\leq$	`\leq`	$\geq$	`\geq`
$\neq$	`\neq`	$\sim$	`\sim`	$\simeq$	`\simeq`
$\approx$	`\approx`	$\cong$	`\cong`	$\equiv$	`\equiv`
$\ll$	`\ll`	$\gg$	`\gg`	$\doteq$	`\doteq`
$\parallel$	`\parallel`	$\perp$	`\perp`	$\in$	`\in`
$\notin$	`\notin`	$\ni$	`\ni`	$\subset$	`\subset`
$\subseteq$	`\subseteq`	$\supset$	`\supset`	$\supseteq$	`\supseteq`
$\succ$	`\succ`	$\prec$	`\prec`	$\succeq$	`\succeq`
$\preceq$	`\preceq`	$\asymp$	`\asymp`	$\sqsubseteq$	`\sqsubseteq`
$\sqsupseteq$	`\sqsupseteq`	$\models$	`\models`	$\vdash$	`\\vdash`
$\dashv$	`\dashv`	$\smile$	`\smile`	$\frown$	`\frown`
$\mid$	`\mid`	$\bowtie$	`\bowtie`	$\propto$	`\propto`
$\lhd$	`\lhd`	$\unlhd$	`\unlhd`	$\rhd$	`\rhd`
$\unrhd$	`\unrhd`	$\sqsubset$	`\sqsubset`	$\sqsupset$	`\sqsupset`
$\bowtie$	`\Join`				

Последние семь из перечисленных команд (от `\lnd` до `\Join`) в LaTeX'е 2_ε определены только в том случае, если вы подключите пакет `\latexsym`.

В следующей таблице собраны стрелки различных видов.

$\to$	`\to`	$\longrightarrow$	`\longrightarrow`	$\Rightarrow$	`\Rightarrow`
$\Longrightarrow$	`\Longrightarrow`	$\hookrightarrow$	`\hookrightarrow`		
$\mapsto$	`\mapsto`	$\longmapsto$	`\longmapsto`	$\leadsto$	`\leadsto`
$\gets$	`\gets`	$\longleftarrow$	`\longleftarrow`	$\leftarrow$	`\leftarrow`
$\Longleftarrow$	`\Longleftarrow`	$\hookleftarrow$	`\hookleftarrow`		
$\leftrightarrow$	`\leftrightarrow`	$\longleftrightarrow$	`\longleftrightarrow`		
$\uparrow$	`\uparrow`	$\Uparrow$	`\Uparrow`		
$\downarrow$	`\downarrow`	$\Downarrow$	`\Downarrow`		
$\updownarrow$	`\updownarrow`	$\Updownarrow$	`\Updownarrow`		
$\nearrow$	`\nearrow`	$\searrow$	`\searrow`		
$\swarrow$	`\swarrow`	$\nwarrow$	`\nwarrow`		
$\leftharpoonup$	`\leftharpoonup`	$\rightharpoonup$	`\rightharpoonup`		
$\rightharpoondown$	`\rightharpoondown`	$\rightleftharpoons$	`\rightleftharpoons`		

Команда `\leadsto` в LaTeX'е будет определена, только если подключить стилевой пакет latexsym.

Из привычных российскому читателю символов в вышеприведенных таблицах нет знаков $\geqslant$ и $\leqslant$, более привычных, чем $\geq$ и $\leq$; кроме того, грече-

ская буква «каппа» лучше смотрится в виде $\varkappa$ (\varkappa), чем в виде κ (\kappa). Эти символы отсутствуют в «классическом» LaTeX'овском наборе; при использовании LaTeX'ом 2_ε они становятся доступными, если подключить в преамбуле стилевой пакет amssymb. Для этого надо после строчки с командой\documentclass написать \usepackage{amssymb}. При условии, что это сделано, можно задавать в математических формулах букву $\varkappa$ командой \varkappa, а символы $\geqslant$ и$\leqslant$ – командами \leqslant и \geqslant соответственно.

5.2.2 Символы пунктуации, акцентирования и интервалы в формулах

Как уже отмечалось, несколько пробелов подряд LaTeX считает за один. Однако возникают ситуации, особенно при наборе формул, когда нужно поставить большой или, наоборот, маленький интервал между символами. В табл. 15 перечислены команды для установки различных горизонтальных интервалов в формулах.

Таблица 15

Команды установки интервалов в формулах

Команда	**Ширина**	**Команда**	**Ширина**
\,	узкий	\:	средний
\;	широкий	\!	отрицательный
\quad	очень широкий	\qquad	самый широкий

Приведем пример.

Исходный текст: `$a\,b\:c\;d\!e \quad f \qquad g $`.

Результат: $a\,b\:c\;d\!e \quad f \qquad g$.

Для пунктуации в формулах используются многоточия, запятые и другие символы, перечень которых дан в табл. 16

Таблица 16

Знаки пунктуации и многоточия

Знак	**Команда**	**Знак**	**Команда**	**Знак**	**Команда**
,	,	;	;	:	\colon
.	\ldotp	·	\cdotp	...	\ldots
⋯	cdots	⋮	\vdots	⋱	\ddots

Подчеркнем, что команды акцентирования в математической моде отличаются от команд в текстовой моде. Например, в математической моде сим-

вол используется для обозначения производной или штриха. Список команд акцентирования приведен в табл. 17

Таблица 17

Команды акцентирования в математической моде

Знак	**Команда**	**Знак**	**Команда**	**Знак**	**Команда**
$\hat{a}$	`\hat{a}`	$\check{a}$	`\check{a}`	$\tilde{a}$	`\tilde{a}`
$\grave{a}$	`\grave{a}`	$\dot{a}$	`\dot{a}`	$\ddot{a}$	`\ddot{a}`
$\acute{a}$	`\acute{a}`	$\breve{a}$	`\breve{a}`	$\bar{a}$	`\bar{a}`
$\vec{a}$	`\ver{a}`	$\widehat{A}$	`\widehat{A}`	$\widetilde{A}$	`\widetilde{A}`

Обращение к этим командам происходит стандартно:

```
$\vec a. \quad \dot{c}. \qquad \widehat{DE}$
```

$\vec{a}. \quad \dot{c}. \qquad \widehat{DE}$

При отсутствии нужного акцента необходимую комбинацию можно подготовить при помощи команды `\stackrel{вверх}{низ}`. Чтобы получить в математической формуле изображение перечеркнутого символа, надо перед соответствующей командой поставить команду `not`:

```
$v \stackrel{\mathrm{def}}{\equiv} 0. \quad \{ x : x \not\in X \}$
```

$v \stackrel{\mathrm{def}}{\equiv} 0, \quad \{x : x \notin X\}$

Для проведения линии над выражением и подчеркивания выражения существуют соответственно команды:

`\overline` и `\underline`

Чтобы провести над выражением левую или правую стрелку, надо обратиться соответственно к командам `\overleftarrow` и `\overrightarrow`

Пример:

```
\begin{displaymath}
\overrightarrow{ABCD} \qquad \underline{l+m+n}
\end{displaymath}
```

$$\overrightarrow{ABCD} \qquad \underline{l+m+n}$$

5.2.3 Степени, индексы, разделители

Для набора степеней и индексов в формулах используются соответственно символы «`^`» и «`_`». Если индексом или степенью является выражение, то его надо заключить в фигурные скобки. Если у символа есть верхний и нижний индексы, то можно указывать в любой последовательности. Для того чтобы верхние и нижние индексы располагались на разных расстояниях от символа, можно использовать «пустые» формулы.

Пример. Исходный текст:

```
\\ $a_1$ \qquad \(x^2\) \qquad
$e^{-\alpha t}$ \qquad $\beta^3_{i+j}$ \\*
\qquad $e^{x^2} \neq{e^x}^2$ \qquad $R_j{}^i_{k l}$
```

Результат:

a_1 x^2 $e^{-\alpha t}$ β^3_{i+j}
$e^{x^2} \neq e^{x2}$ R_{jkl}^{i}

Квадратный корень вводится как `\sqrt`, а корень степени `n` генерируется командой `\sqrt[n]`. Высота и ширина знака корня определяются системой LaTeX автоматически. Чтобы поставить только знак корня, не продолжая его над последующими символами, используется команда `surd`.

Часто размеры знака корня у соседних символов оказываются различными. Для того чтобы таких явлений не было, используются пустые (невидимые) символы - «страты», имеющие определенную ширину и высоту:

- `\mathstrut` – невидимый символ, имеющий высоту круглой скобки;
- `\vphantom{math}` – символ, имеющий высоту формулы `math`;
- `\phantom{math}` – пробел ширины выражения `math`;

Приведем демонстрационный пример. Исходный текст:

```
$\sqrt{d^2+x}+\sqrt{y}$ \\*
$\sqrt{\mathstrut d^2+x}+\sqrt{\mathstrut y}$ \\*
$\sqrt[3]{2} \: \surd (x^2+y^2)$
```

Результат:

$\sqrt{d^2+x}+\sqrt{y}$
$\sqrt{\mathstrut d^2+x}+\sqrt{\mathstrut y}$
$\sqrt[3]{2} \: \surd (x^2+y^2)$

Дроби можно писать в одну строку при помощи символа наклонной черты «/», а для дробей на нескольких уровнях применяется команда: `\frac{числитель}{знаменатель}`.

Приведем пример. Исходный текст:

```
Дробь в тексте. \\
Вариант 1: $(x+1)/x$\\
Вариант 2:  $\frac{x+1}{x}$ \\
А вот как она выглядит в выключной формуле:
\begin{displaymath}
(x+1)/x, \qquad \frac{x+1}{x}.
\end{displaymath}
```

Результат:
Дробь в тексте.
Вариант 1: $(x+1)/x$
Вариант 2: $\frac{x+1}{x}$
А вот как она выглядит в выключной формуле:

$$(x+1)/x, \qquad \frac{x+1}{x}.$$

5.3 Математические операторы

В следующей таблице собраны команды для воспроизведения названий математических операторов наподобие sin, log и т.п., обозначаемых последовательностью букв, набираемых прямым шрифтом. Любой из этих операторов можно снабдить верхним и/или нижним индексом.

Таблица 18

Математические операторы

Оператор	**Команда**	**Оператор**	**Команда**	**Оператор**	**Команда**
log	`\log`	lg	`\lg`	ln	`\ln`
arg	`\arg`	ker	`\ker`	dim	`\dim`
hom	`\hom`	deg	`\deg`	exp	`\exp`
sin	`\sin`	arcsin	`\arcsin`	cos	`\cos`
arccos	`\arccos`	tan	`\tan`	arctan	`\arctan`
cot	`\cot`	sec	`\sec`	csc	`\csc`
sinh	`\sinh`	cosh	`\cosh`		
coth	`\coth`	tanh	`\tanh`		

В этой таблице обозначения tan, arctan и т.д. – не что иное, как принятые в англоязычной литературе обозначения для тангенса, арктангенса и т.д. В отечественной литературе, однако же, принято обозначать tg, ctg и т.д.

Так как в стандартном комплекте TEX'а или LATEX'а команд для этого нет, их приходится при необходимости определять самому. Это просто: в преамбуле документа надо написать следующую команду:
`\newcommand{\tg}{\mathop{\rm tg}\nolimits}`.

После этого команда `\tg` будет создавать в математической формуле запись tg с правильными пробелами вокруг нее. Другие команды такого типа определяются аналогично, надо только вместо tg написать то название функции (скажем, arctg), которое должно появиться на печати. Если вы получили LATEX вместе с русификацией, то не исключено, что в ней уже определены команды для принятых в России обозначений тангенса, арктангенса и т.п.

Описанный выше способ определения команд является частным случаем существующей в LaTeX'е конструкции для определения новых команд (подробнее см. раздел 6).

5.4 Операции с пределами

Теперь обсудим, как можно было бы получить, скажем, формулу

$$\sum_{i=1}^{n} n^2 = \frac{n(n+1)(2n+1)}{6}$$

с дополнительными элементами над и под знаком операции суммирования – $\sum$. В данной формуле эти элементы называются пределами суммирования, поэтому в TeX'нической терминологии записи над и под знаком операции принято называть пределами (по-английски limits). В исходном тексте пределы обозначаются точно так же, как индексы; имея в виду, что знак суммы генерируется командой `\sum`, получаем, что вышеназванную формулу можно получить так:

```
$$\sum_{i=1}^n {n^2}=\frac{n(n+1)(2n+1)}{6}$$
```

В этом примере существенно, что формула была выключной; во внутритекстовой формуле пределы печатаются на тех же местах, что их индексы:

Тот факт, что $\sum_{i=1}^{n}(2n-1)=n^2$, следует из формулы для суммы арифметической прогрессии	`Тот факт, что $\sum_{i=1}^n (2n-1)=n^2$, следует из формулы для суммы арифметической прогрессии`

Можно добиться, чтобы пределы были сверху и снизу знака оператора. Рассмотрим это на примере еще одной математической операции, для которой требуются пределы, – это интеграл. В LaTeX'е есть команды `\int` для обычного знака интеграла $\int$ и `\oint` для знака «контурного интеграла» $\oint$. При этом, для экономии места, пределы интегрирования помещаются не сверху и снизу от знаков интеграла, а по бокам (даже в выключных формулах): `$$\int_0^1x^2\,dx=1/3 $$`

$$\int_0^1 x^2\,dx = 1/3.$$

Если, тем не менее, необходимо, чтобы пределы интегрирования стояли над и под знаком интеграла, то надо непосредственно после `\int` записать команду `\limits`, а уже после нее - обозначения для пределов интегрирования:

```
$$\int\limits_0^1x^2\,dx=1/3$$
```

$$\int\limits_0^1 x^2\,dx=1/3.$$

Тот же прием с командой `\limits` можно применить, если хочется, чтобы во внутритекстовой формуле пределы у оператора стояли над и под ним, а не сбоку.

Если, с другой стороны, надо, чтобы пределы у какого-либо оператора стояли не над и под знаком оператора, а сбоку, то после команды для знака оператора надо записать команду `\nolimits`, а уже после нее – обозначения для «пределов»:

```
$$\prod\nolimits_{i=1}^ni=n!$$
```

$$\prod\nolimits_{i=1}^n i=n!$$

Вот список операторов, ведущих себя так же, как `\sum` и `\int`:

$\sum$	`\sum`	$\prod$	`\prod`	$\bigcup$	`\bigcup`
$\bigcap$	`\bigcap`	$\coprod$	`\coprod`	$\bigoplus$	`\bigoplus`
$\bigotimes$	`\bigotimes`	$\bigodot$	`\bigodot`	$\bigvee$	`\bigvee`
$\bigwedge$	`\bigwedge`	$\biguplus$	`\biguplus`	$\bigsqcup$	`\bigsqcup`
$\lim$	`\lim`	$\limsup$	`\limsup`	$\liminf$	`\liminf`
$\max$	`\max`	$\min$	`\min`	$\sup$	`\sup`
$\inf$	`\inf`	$\det$	`\det`	$\Pr$	`\Pr`
$\gcd$	`\gcd`				

Все обозначения из этой таблицы употребительны в отечественной литературе, за исключением `\gcd` для наибольшего общего делителя (у нас его иногда обозначают НОД) и `\Pr` для вероятности, обычно обозначаемой P.

Бывает нужно, чтобы один из пределов состоял из нескольких строк, тогда используется команда `\substack{line1\\line2}` из пакета `amsmath`. Пример:

$$\prod_{\substack{1\le i\le n\\ 1\le j\le m}} a_{ij}$$

```
$$
\prod_{\substack{1 \le i \le n\\
1 \le j \le m}} a_{ij}
$$
```

5.5 Применение скобок

Если перед одной скобкой стоит `\left`, а перед другой скобкой стоит `\right`, то на печати размер этих скобок будет соответствовать высоте фрагмента формулы, заключенного между `\left и \right`.

Конструкция с `\left и \right` применима не только к круглым скобкам. В следующей табл. 19 перечислены скобки и некоторые другие символы, которые с помощью `\left и \right` автоматически принимают нужный размер. TEXнический термин для таких символов – ограничители (по-английски delimiters).

Таблица 19

Символы-ограничители

Символ	**Команда**	**Символ**	**Команда**	**Символ**	**Команда**
$($	`(`	$)$	`)`	$[$	`[`
$]$	`]`	$\{$	`\{`	$\}$	`\}`
$\lfloor$	`\lfloor`	$\rfloor$	`\rfloor`	$\lceil$	`\lceil`
$\rceil$	`\rceil`	$\langle$	`\langle`	$\rangle$	`\rangle`
$\vert$	`|`	$\Vert$	`\\|`	$/$	`/`
$\backslash$	`\backslash`				

Вместо `\left\langle` можно писать `\left<` , и аналогичным образом вместо `\right\rangle` можно писать `\right>` (однако же $<$ – это не `\langle`!).

Вместе с каждой командой `\left` в формуле должна присутствовать соответствующая ей команда `\right`, в противном случае TEX выдаст сообщение об ошибке. Вместе с тем TEX вовсе не требует, чтобы «ограничители» (например, скобки) при командах `\left и \right` были расположены сколько-нибудь осмысленно с математической точки зрения: вы вполне можете написать что-нибудь вроде `\left(...\right]` или даже, вопреки смыслу слов left и right, `\left) ..\right(`.

Вместо «ограничителя» после команды `\left или \right` можно поставить точку. На месте этой точки ничего не напечатается, а другой «ограничитель» будет необходимого размера. Вот пример того, как можно использовать этот прием. Таким способом можно создать косую дробную черту увеличенного размера (символ / также является «ограничителем» – см. табл. 19):

$$M(f) = \left(\int\limits_a^b f(x)\,dx \right) \Big/ (b-a)$$

```
$$
M(f)=\left.\left(
\int\limits_a^b
f(x)\,dx
\right)
\right/(b-a)
$$
```

5.6 Шрифты и текст в формулах

Размер символов в формулах и их начертание в большинстве случаев регулируется LaTeX'ом автоматически. Но иногда необходимо указать нужные параметры вручную. Следует помнить, что в формулах используются четыре размера шрифта:

```
${\displaystyle Text}, {\textstyle Text},$\\
${\scriptstyle Text}, {\scriptscriptstyle Text}$
```

Результат применения команд изменения шрифта к слову Text:
$Text, Text, {\scriptstyle Text}, {\scriptscriptstyle Text}$.

Данные виды шрифтов имеют относительные размеры, автоматически рассчитываемые по величине базового шрифта tex-документа. Но их можно задать явно в преамбуле с помощью команды:
`DeclareMathSizes{display}{text}{script}{scriptscript}`.

В каждой фигурной скобке необходимо указать размер соответствующего шрифта в порядке, указанном выше, в любых известных TeX'у единицах измерения.

Продемонстрируем применение команд изменения шрифта. Для сложных дробей с несколькими уровнями вложенности возникает проблема с размерами символов, которую можно разрешить, управляя стилями формулы:
Так выглядит формула, набранная обычным способом:

```
\[ \frac{1}{x+ \frac{1}{x+\frac{1}{x+\frac{x}{x+1}}}}\]
```

$$\frac{1}{x+\frac{1}{x+\frac{1}{x+\frac{x}{x+1}}}}.$$

А это усовершенствованная формула:

```
\[ \frac{1}{x+\displaystyle
\frac{1}{x+\displaystyle\frac{1}{x+\displaystyle
\frac{x}{x+1}}}}\] \\
```

$$\frac{1}{x+\dfrac{1}{x+\dfrac{1}{x+\dfrac{x}{x+1}}}}.$$

В следующей табл. 20 представлены разные команды изменения начертания шрифта в формулах.

Таблица 20

Команды изменения шрифтов в формулах

Команда	**Пример**	**Необходимый пакет**
`mathrm{ABCdef}`	ABCdef	
`mathbf{ABCdef}`	$\mathbf{ABCdef}$	
`mathit{ABCdef}`	ABCdef	
`mathnormal{ABCdef}`	$\mathnormal{ABCdef}$	
`mathcal{ABC}`	$\mathcal{ABC}$	
`mathfrak{ABCdef}`	$\mathfrak{ABCdef}$	Пакет `eufrak`
`mathbb{ABC}`	$\mathbb{ABC}$	Пакет `amsfonts` или `amssymb`

Обычно LaTeX игнорирует любые текстовые строки в формулах, не являющиеся именами операторов и переменных. Но вставить фрагмент текста в математическую формулу все-таки можно. Для этого используется команда `\mbox`, оформленная по описанным в разделе 3.5 правилам:

```
\begin{displaymath}
\forall \: x \: \exists \: \delta \: \mbox{такое, что}
\left\|x-\delta\right\| \leqslant \varepsilon
\end{displaymath}
```

$$\forall\, x\, \exists\, \delta \text{ такое, что } \|x - \delta\| \leqslant \varepsilon$$

5.7 Один элемент над другим

В разделах 5.2.2 и 5.4 говорилось о частных случаях размещения одного элемента над другим (пределы, дроби и акценты). Сейчас рассмотрим общий случай.

Для набора столбцов из двух элементов, можно пользоваться командами: `{... \choose ...}` или `{... \atop ...}`

Первая команда отличается от второй тем, что столбец заключается в скобки.

Приведем пример. Исходный текст:

```
Иллюстрация двухстрочных элементов:
\[ {x+1 \choose x}, \quad {x+1 \atop x} \].
```

Результат:

Иллюстрация двухстрочных элементов:

$$\binom{x+1}{x}, \quad \begin{matrix} x+1 \\ x \end{matrix}.$$

Чтобы нарисовать горизонтальную фигурную скобку под выражением (а после сделать подпись под этой скобкой), надо воспользоваться командой `\underbrace`. Аргумент этой команды – тот фрагмент формулы, под которым

надо провести скобку; подпись под скобкой, если она нужна, оформляется как нижний индекс. Например, такая формула:

$$\underbrace{1+3+5+7+\cdots+2n-1}_{n \text{ слагаемых}}=n^2$$

получается следующим образом:

```
$$
\underbrace{1+3+5+7+\cdots+2n-1}_{\mbox{$n$ слагаемых}}=n^2
$$
```

Горизонтальная фигурная скобка над фрагментом формулы генерируется командой `\overbrace`, надпись над ней оформляется как верхний индекс. В одной формуле могут присутствовать горизонтальные фигурные скобки как над, так и под фрагментом формулы:

$$\overbrace{\underbrace{a+b+\cdots+z}_{26}+1+\cdots+10}^{36}$$

```
$$
\overbrace{\underbrace{
a+b+\cdots+z
}_{26}+1+
\cdots+10}^{36}
$$
```

5.8 Набор матриц

Чтобы набрать с помощью LaTeX'а матрицу, надо воспользоваться окружением array. Чтобы понять, как это окружение работает, разберем такой пример:

$$\begin{array}{cccc}
a_{11} & a_{12} & \ldots & a_{1n} \\
a_{21} & a_{22} & \ldots & a_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
a_{n1} & a_{n2} & \ldots & a_{nn}
\end{array}$$

```
$$
\begin{array}{cccc}
a_{11}&a_{12}
&\ldots&a_{1n}\\
a_{21}&a_{22}
&\ldots&a_{2n}\\
\vdots&\vdots
&\ddots&\vdots\\
a_{n1}&a_{n2}
&\ldots&a_{nn}
\end{array}
$$
```

Посмотрим, как устроен исходный текст, давший на печати эту матрицу. Структура матрицы похожа на устройство таблицы `tabular`, рассмотренной в разделе 4.3.3: строки матрицы разделяются с помощью команды \\ (последнюю строку можно не заканчивать командой \\), а элементы одной строки отделяются друг от друга с помощью символа `&`. После `\begin{array}`, открывающего окружение, в фигурных скобках описывается «преамбула матрицы», определяющая количество столбцов и выравнивание элементов по горизонтали.

В рассматриваемом примере четыре буквы cccc. Это значит, что в матрице 4 столбца (по букве на столбец) и что содержимое каждого из этих столбцов выравнивается по центру. Кроме c, в преамбуле может стоять буква l – выравнивание по левому краю, или r – выравнивание по правому краю.

Матрице не хватает еще скобок; чтобы их создать, надо написать `\left(` перед `\begin{array}` и `\right)` после `\end{array}` (см. раздел 5.5).

Окружение array можно использовать не только для матриц: это окружение просто создает массивы, состоящие из строк и столбцов. Вот, например, как можно напечатать треугольник Паскаля:

$$
\begin{array}{ccccccccccc}
 & & & & 1 & & 1 & & & & \\
 & & & 1 & & 2 & & 1 & & & \\
 & & 1 & & 3 & & 3 & & 1 & & \\
 & 1 & & 4 & & 6 & & 4 & & 1 & \\
1 & & 5 & & 10 & & 10 & & 5 & & 1
\end{array}
$$

Исходный текст для него выглядит так:

```
$$
\begin{array}{ccccccccccc}
&&&& 1 && 1
&&& 1 && 2 && 1
&& 1 && 3 && 3 && 1
& 1 && 4 && 6 && 4 && 1
1 && 5 && 10 && 10 && 5 && 1
\end{array}
$$
```

Если в таблице отсутствует какой-то элемент, то в соответствующей ячейке, ограниченной `&`, нужно просто ничего не писать (или оставить сколько угодно пробелов). Если после того, что вы написали в строке, до конца строки идут только пустые ячейки, то можно не дописывать до конца значки `&`, а сразу написать \\.

Разберем еще один пример, типичный при работе в LaTeX'е 2.09: верстку системы уравнений с помощью окружения array.

$$\left\{ \begin{array}{rcl} x^2+y^2 &=& 7 \\ x+y &=& 3 \end{array} \right.$$

```
$$
\left\{
\begin{array}{rcl}
x^2+y^2&\!=\!&7\\ x+y &\!=\!&3\\
\end{array}
\right.
$$
```

Мы отвели по одному столбцу на левую часть каждого уравнения, на знак равенства и на правую часть. При этом левые части выровнены по правому краю, а правые части – по левому краю, а знак равенства расположен по центру колонки.

Для создания фигурной скобки, охватывающей всю систему слева, мы воспользовались командами `\left\{` и `\right.`, причем при команде `\right` стоит «пустой ограничитель» – точка (см. разд. 5.5). Обратите внимание на отрицательные пробелы `\!` с двух сторон знака равенства, которые уменьшают пробелы (отбивки) до размеров, допустимых типографскими правилами.

6 Программирование в системе LaTeX

6.1 Создание собственных команд и окружений

В пакете LaTeX имеется довольно мощный набор команд, которые позволяют значительно облегчить верстку как простых, так и сложных документов. Можно переопределять уже существующие команды и окружения, а можно писать новые. Простым и эффективным примером служит определение новых команд для сокращения стандартных имен.

Для определения новых команд в системе LaTeX имеется команда
`\newcommand{name}[num]{definition}`,
где обязательными являются первый и третий аргументы, которые соответственно задают имя и содержание новой команды. Необязательный аргумент num определяет количество аргументов команды, которое может изменяться от 1 до 9. Аргументы в тексте определения команды обозначаются двумя символами: «#» и номером аргумента.

Для переопределения существующих команд предназначена команда
`\renewcommand{name}[num]{definition}`.
Смысл параметров аналогичен описанным для команды `\newcommand`.

Удобно для часто встречающегося сочетания слов или формулы определить команду, генерирующую такой текст. Приведем примеры.
Исходный текст:

```
\newcommand{\o}{ортогональн}\\
Набор {\o}ых полиномов удовлетворяет условию {\o}ости.
```

Результат:

Набор ортогональных полиномов удовлетворяет условию ортогональности.

Исходный текст:

```
$\omega$ - полнота. \renewcommand{\omega}{Омега} \omega - полнота.
```

Результат:

ω - полнота. \renewcommand{\omega}{Омега} \omega - полнота.

Исходный текст:

```
\newcommand{\Func}[2]{\frac{\sqrt{#2}}{\sqrt[3]{{#1}^4+({#2}-6)}}}\\
$\Func{1}{2}$, $\Func{a+b}{c+d}$
```

Результат:

$$\frac{\sqrt{2}}{\sqrt[3]{(1)^4+(2-6)}}, \frac{\sqrt{c+d}}{\sqrt[3]{(a+b)^4+(c+d-6)}}$$

С помощью данных команд можно определять новые операторы, как было показано в разделе 5.3. Но стоить помнить, что в пакете `amsmath` имеется специальная команда для этого: `\DeclareMathOperator\{name}{definition}`. Аргументы команды аналогичны соответствующим аргументам `\newcommand`.

Если же оператор встречается в тексте только один-два раза, то можно внутри формулы использовать команду `\operatorname{\name}` для оформления аргумента name по правилам оформления операторов в формулах.

Для определения нового окружения и изменения уже существующего используются соответственно команды:
`\newenvirorment{name}[num]{before}{after}`,
`\renewenvironment{name}[num]{before}{after}`.
Здесь обязательный аргумент name задает имя окружения, а необязательный аргумент num определяет число формальных аргументов окружения. Аргумент before здесь обозначает набор команд, выполнение которых предшествует анализу текста находящегося внутри окружения, а after – команды, выполняющиеся по завершении анализа. Отметим, что формальные аргументы окружения могут находиться только в группе команд before.

В пакете `amsmath` есть специальная команда для определения нового окружения типа теорема: `\newtheorem\{name}{definition}`. При помощи данной команды можно задать свое оформление и организовать автоматическую нумерацию лемм, утверждений, следствий, примеров и пр.

6.2 Работа со счетчиками

Автоматическую нумерацию каких-либо частей документа можно легко организовывать при помощи счетчиков. Счетчик – это специальная переменная, принимающая целые значения. Счетчику можно присваивать значение

и изменять его, выводить его значение на печать и организовывать с его помощью автоматическую генерацию ссылок.

Мы уже касались стандартных счетчиков в разделе 4.2. Рассмотрим вопрос создания новых счетчиков и работу с их значениями.

Каждый счетчик имеет свое имя и создается с помощью команды:
`\newcounter{new}{old}`,
где new – имя вновь организуемого счетчика, а old – имя уже существующего счетчика, которому будет подчинен вновь организуемый счетчик. Второй аргумент этой команды не является обязательным.

Для изменения значения существующего счетчика используются ранее упомянутые команды: `\setcounter{name}{num}`, `\addtocounter{name}{num}`. Напомним, что первая команда устанавливает счетчику с именем name значение num, а вторая увеличивает значение счетчика name на величину num.

Существуют еще две команды, которые увеличивают значение счетчика на единицу и обнуляют все подчиненные счетчики: `\refstepcounter{count}`, `\stepcounter{count}`. Вторая команда применяется реже, так как с ее помощью нельзя организовать автоматические ссылки.

Как стандартные, так и созданные пользователем счетчики могут выводиться арабскими, римскими цифрами и символами латинского алфавита. По умолчанию значение счетчика выводится арабскими цифрами, изменить вывод можно следующими командами:

\arabic{count} – для арабских цифр;

\roman{count} и \Roman{count} – для малых и больших римских цифр;

\alph{count} и \Alph{count} – для малых и больших латинских букв.

Приведем пример, включающий определение нового счетчика и организацию автоматических ссылок:

```
\newcounter{z}\\
\newcommand{\zdch}
{\par\textbf{Задача \addtocounter{z}{1}\arabic{z}. }}\\
\zdch Сформулировать \ldots \\
\zdch Доказать \ldots\\
\zdch Применить \ldots \\
```

Результат:

Задача 1. Сформулировать ...

Задача 2. Доказать ...

Задача 3. Применить ...

7 Компиляция файла и обработка ошибок

Для компиляции tex-документа в dvi- или pdf-формат нужно выбрать одну из программ-компиляторов через соответствующие инструменты специализированного редактора tex-файлов (TeXworks, WinEdt, др.) и нажать на кнопку компиляции. А можно просто через командную строку ввести команду вида <programm> <file.tex>, где <programm> – название программы-компилятора, а <file.tex> – имя исходного файла. Например, pdflatex main.tex.

Приведем список наиболее известных компиляторов:

tex – простейший компилятор, берет TEX-файл и создает DVI-файл;

pdftex – берет TEX-файл и создает PDF-файл;

latex – наиболее используемый: берет LATEX-файл и создает DVI-файл;

pdflatex – берет LATEX-файл и создает PDF-файл;

dvips – конвертирует DVI-файл в PostScript;

dvipdf – конвертирует DVI-файл в PDF;

dvipdfm – улучшенная (с некоторых точек зрения) версия предыдущей программы.

При наборе текстов неизбежно возникают ошибки. При компиляции документа все сообщения об ошибках и предупреждения о неточностях показываются на экране и, кроме того, записываются пакетом в файл протокола (расширение .log). Предупреждения не вызывают прерывания процесса компиляции, а при обнаружении синтаксической ошибки компиляция останавливается и на экран выводится сообщение об ошибке.

При компиляции LATEX-файла возможны ошибки двух типов: ошибки пакета LATEX и ошибки языка TEX, то есть ошибки более низкого уровня. Если компилятор обнаруживает ошибку, то печатает ее тип (LATEX-error или

TEX-error), номер строки исходного файла, в которой, по мнению пакета, находится ошибка, и само ее содержание. Сообщение заканчивается знаком вопроса. При этом компиляция приостанавливается и пакет переходит в режим ожидания реакции пользователя.

Ошибку можно проигнорировать, нажав клавишу Enter. В этом случае LATEX сам попробует ее исправить по своему разумению. Набрав <h> и нажав Enter, можно посмотреть информацию об ошибке на английском языке.

Если ошибок много, можно при остановке компиляции ввести с клавиатуры <s> и «ввод», тогда при обнаружении дальнейших ошибок компиляция прерываться не будет (TEX будет обрабатывать ошибки так, как если бы вы все время нажимали на «ввод»), по экрану пронесутся сообщения об ошибках, а затем вы сможете их изучить, просмотрев log-файл.

Вместо <s> можно ввести <r> и «ввод»: результат будет такой же, как если бы вы сказали s, с той разницей, что в случае, когда аргументом команды `\input` служит несуществующий файл, никаких вопросов задаваться не будет, а компиляция просто прервется.

Можно набрать <q> и «ввод»: результат будет такой же, как от r, с той разницей, что на экран не будет выдаваться вообще ничего (в log-файл все будет записано).

Ошибки можно исправлять в диалоговом режиме. Для этого при остановке необходимо ввести <i> и «ввод», после чего появится предложение <insert>, куда нужно ввести исправленную команду. Но затем все равно необходимо исправить ошибку в исходном файле.

А можно прекратить компиляцию исходного файла, нажав клавиши <x> и «ввод».

Наконец, режимы реакции на ошибки, задаваемые с клавиатуры с помощью клавиш s, r или q, можно задать прямо в файле, написав в преамбуле одну из перечисленных ниже команд:

1. Команда `\scrollmode` равносильна нажатию <s>;
2. Команда `\nonstopmode` равносильна нажатию <r>;
3. Команда `\batchmode` равносильна нажатию <q>.

Список литературы

1. Говорухин В., Цибулин Б. Компьютер в математическом исследовании: Maple, MATLAB, LaTeX.[Текст]/Говорухин В., Цибулин Б. – СПб.: Питер, 2001. – 624 с.

2. Гуссенс М., Миттельбах Ф., Самарин А. Путеводитель по пакету LaTeX и его расширению LaTeX 2ε.[Текст]/Гуссенс М., Миттельбах Ф., Самарин А. – М.: Мир, 1999. – 606 с.

3. Львовский С.М. Работа в системе LaTeX [Текст]/Львовский С.М. – М.Национальный Открытый Университет «ИНТУИТ» , 2007. – 465 с.

4. Столяров А.В. Сверстай диплом красиво: LATEX за три дня [Текст]/ Столяров А.В. – М.: МАКС Пресс, 2010. – 100 с.

Чебарыков Михаил Сергеевич

ОСНОВЫ РАБОТЫ В СИСТЕМЕ LaTeX

Учебное пособие для студентов
направления "Информатика и вычислительная техника"

Редактор Е.Ф. Изотова

Подписано к печати 29.12.14. Формат 60x84 1/16.
Усл. печ. л. 3,06. Тираж 35 экз. Зак. № 141366. Рег. № 222
Отпечатано в ИТО Рубцовского индустриального института.
658207, г. Рубцовск, ул. Тракторная, 2/6.