LaTeX: краткое описание

Hubert Partl[*]
Elisabeth Schlegl[†]
Irene Hyna[*]
Сокращенный пересказ с немецкого А. Шеня[‡]

Version 3
17. Mai 1990
Русский вариант: сентябрь 1993

LaTeX [1, 2] — система подготовки текстов типографского качества с формулами, построенная на основе системы TeX [4, 5].

Данный текст кратко описывает систему LaTeX Version 2.09; подробнее смотри в *LaTeX-Manual* [1].

Система LaTeX реализована на разных компьютерах. Особенности реализации должны быть отражены в *Local Guide* [3].

---

[*] EDV-Zentrum der Technischen Universität Wien, Abt. Digitalrechenanlage

[†] EDV-Zentrum der Karl-Franzens-Universität Graz

[‡] ИППИ РАН, 57-ая школа, Независимый Московский университет

Данный текст подготовлен с помощью системы LaTeX. Текст немецкого оригинала может быть получен по электронной почте у `listserv@dhdurz1.bitnet` *(Heidelberg). Русский вариант (в виде файла) спрашивайте у знакомых — он, как и немецкий оригинал, свободно распространяется и есть, например, в 43-й и 57-ой московских школах. К нему добавлен раздел «Вокруг* TeX*а», написанный А. Шенем (shen@sch57.msk.su, shen@math.imu.msk.su). Все материалы, использованные при подготовке данной книги — текстовый редактор, комплект em*TeX *Эберхарда Маттеса, русские шрифты к нему, текст книги в виде файла — являются свободно распространяемыми.*

Содержание

Список иллюстраций

Список таблиц

1 Общие замечания

1.1 Название

1.1.1 TeX

Программа TeX (произносится “тех”, пишется также “TeX”) написана Дональдом Кнутом [4, 5] и предназначена для набора текстов с формулами.

1.1.2 LaTeX

LaTeX (произносится “латех” или “лэйтех”, пишется также “LaTeX”) — это надстройка над TeX (пакет макроопределений). Его автор — Leslie Lamport [1, 2].

1.2 Основные идеи

1.2.1 Автор, техред и наборщик

Получив от автора рукопись, технический редактор размечает ее для набора. В этой традиционной схеме LaTeX играет роль техреда, а TeX — наборщика.

В традиционной схеме формат книги определялся редактором, а автор мог лишь высказывать свои пожелания. В нынешних системах экранного редактирования автор видит на экране, как будет выглядеть набираемый им текст. LaTeX не относится к их числу и ближе к традиционной схеме: автор, готовя исходный файл, указывает лишь логическую структуру текста. Как он будет выглядеть, станет ясно лишь после пробной печати.

1.2.2 Выбор оформления

Выбор оформления книги — сложное дело. Системы экранного редактирования (WYSIWYG) возлагают ответственность на автора, что не всегда желательно. LaTeX позволяет автору задать логическую структуру текста (разбиение на главы, разделы и т.п.) и оставляет за собой конкретные детали оформления.

1.2.3 Достоинства и недостатки

По сравнению с другими системами достоинствами системы LaTeX являются:

- Несколько готовых форматов позволяют получить напечатанный текст “совсем как в книге”.
- Особенно хорошо выглядят формулы.
- Указав с помощью простых средств логическую структуру текста, автор может не вникать в детали оформления.
- Сноски, таблицы, оглавление и т.п. получаются почти что сами собой.

LaTeX имеет и недостатки:

- Он работает медленно и занимает много памяти.
- Полученный результат нельзя напечатать на дешевом принтере.
- Хотя параметры оформления менять легко, создание принципиально нового оформления (нового “стиля”) — весьма непростое дело.

1.3 LaTeX в России

Использование системы LaTeX в России прежде всего требует адаптации ее к русским буквам. Кроме того, типографские традиции в разных странах различны (например, в англоязычных странах используют кавычки-“апострофы”, в то время как в России стандартом являются «елочки» и „лапки“). Различаются и обозначения (русский tg — английский tan и т.п.). Многое из этого учтено в написанном С.М.Львовским файле `russcorr.sty`, позволяющем приспособить LaTeX к российским традициям. Мы, однако, про это ничего не пишем, поскольку С.М.Львовский сейчас пишет более подробное руководство по системе LaTeX, которое — будем надеяться — вскоре будет готово и будет (свободно) распространяться вместе с «русифицированным» комплектом LaTeX.

1.4 Исходные файлы

Исходный файл для системы LaTeX — это ASCII-файл, создаваемый любым текстовым редактором. Он содержит указания системе LaTeX, касающиеся размещения текста.

1.4.1 Пробелы

Пробелы разных видов в исходном тексте (в том числе концы строк) не различаются. Несколько последовательных пробелов равносильны одному, за одним исключением: пустая строка (т.е. два перевода строки подряд) означает конец абзаца. Несколько пустых строк неотличимы от одной. Интервалы между словами и строками можно регулировать специальными командами.

1.4.2 Спецзнаки

Следующие символы имеют особый смысл и не могут встречаться в тексте “просто так”:

```
$ & % # _ { }  ~  ^ "  \  | < >
```

Некоторые из них могут быть набраны с помощью обратной косой черты \ (backslash):

$ & % # _ { }

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

Остальные символы (и многие другие) могут быть набраны с помощью специальных команд.

1.4.3 LaTeX-команды

Большинство LaTeX-команд имеет такой вид: они начинаются с обратной косой черты (\) за которой следуют либо несколько букв (до первой не-буквы) либо ровно один спецзнак или цифра. Большие и маленькие буквы различаются. Пробелы после команды из букв, независимо от их числа, игнорируются. Если после команды в тексте должен стоять пробел, то надо в ее конце поставить `{}`, за которым и идет этот пробел. Можно также использовать знак \, за которым идет пробел, как в одном из примеров.

Сегодня 35 мая 1987. Или: Сегодня 35 мая 1987. Неверно: Утром 35 мая 1987шел дождь. Верно: Утром 35 мая 1987 было ясно. Или: Утром 35 мая 1987 шел снег.

```
Сегодня \today. Или:
Сегодня \today . Неверно:
 Утром \today шел дождь.
Верно: Утром \today{}
было ясно. Или: Утром
\today\ шел снег.
```

Некоторые команды имеют параметры, указываемые вслед за командой в фигурных скобках. Необязательные параметры некоторых команд указываются в квадратных скобках. Некоторые команды имеют варианты, в которых после имени команды ставится звездочка.

Фигурные скобки используются также для указания “области действия”. Действие команд внутри фигурных скобок или внутри одного из “окружений” не выходит за их границы. В приведенном выше примере `{}` представляет собой пустую группу, единственное назначение которой — ограничивать команду `today`.

1.4.4 Комментарии

Все, что стоит после процента (`%`) и до конца строки, является комментарием и игнорируется. (Здесь автор может делать пометки не для печати.)

Это пример.

```
Это % глупый
% Лучше: поучительный <----
пример.
```

1.4.5 Структура входного файла

Первой командой LaTeX-файла должна быть команда

```
\documentstyle
```

(смотри раздел 1.5.1). Далее следуют определения, относящиеся ко всему тексту. Команда

```
\begin{document}
```

начинает набираемый текст. За ней следует текст со всеми входящими в него LaTeX-командами, который завершается командой

```
\end{document}
```

(Все дальнейшее LaTeX игнорирует.)

На рис. 1 показан минимальный LaTeX-файл. Более сложный пример показан на рис. 2.

```
\documentstyle{article}
\begin{document}
Лучше меньше, да лучше.
\end{document}
```

Рис. 1: Минимальный LaTeX-файл

1.5 Оформление текста

1.5.1 “Стиль”

В начале входного файла должен быть задан “стиль” документа с помощью конструкции

`\documentstyle[`*варианты*`]{`*стиль*`}`

Внутри фигурных скобок должен быть указан один из возможных стилей, перечисленных в таблице 1.

В квадратных скобках могут стоять (а могут и не стоять) дополнительные параметры, задающие тот или иной вариант выбранного стиля. Наиболее важные параметры стандартных стилей указаны в таблице 2.

Пример: исходный файл для немецкого варианта этого описания начинается словами

```
\documentstyle[11pt]{article}
\author{H.~Partl}
\title{О длинном и коротком}
\setlength{\parindent}{0pt}
\setlength{\parskip}{5pt plus 2pt minus 1pt}
\frenchspacing
\sloppy

\begin{document}
\maketitle
\begin{abstract}
Пример статьи на русском языке.
\end{abstract}
\tableofcontents

\section{Начало}

Здесь -- начало моего эссе\dots

\section{Конец}

\dots\ а здесь его конец.

\end{document}
```

Рис. 2: Набросок статьи

Таблица 1: Стили

`article`	статьи в журналах и др.
`report`	отчеты из нескольких глав, диссертации и т.п.
`book`	книги
`letter`	письма

Таблица 2: Варианты различных стилей

`11pt`	11 ( )
`12pt`	12 ( ) Когда не указан ни один из этих вариантов, по умолчанию используется шрифт кегля 10 (стандартный размер книжного шрифта)
`fleqn`	уравнения помещаются слева (а не в центре)
`leqno`	номера уравнений слева (а не справа)
`titlepage`	отдельный титульный лист (относится только к стилю `article`)
`twocolumn`	печать в две колонки
`twoside`	печать на обеих сторонах бумаги (разные поля на четных и нечетных страницах)

```
\documentstyle[11pt,twoside]{article}
```

(Русский вариант набран кеглем 10, так что 11pt опущено.)

1.5.2 Формат страницы

Формат страницы (при желании) можно изменить командой

```
\pagestyle{стиль}
```

Указав в качестве аргумента этой команды `plain` (или ничего не указав), мы получим номер страницы в нижней строке. Указав `headings`, мы получим колонтитулы и номер страницы в верхней строке. Наконец, `empty` ничего не печатает снизу и сверху.

В *LaTeX-Manual* [1] описано, как воздействуют команды `\thispagestyle`, `\pagenumbering`, `\twocolumn` и `\onecolumn` на формат страницы.

2 Набор текста

2.1 Разбиение на строки и страницы

2.1.1 Выравнивание

Обычно текст выравнивается по правому краю. LaTeX делает это автоматически, при необходимости перенося слова.

Слова и предложения разделяются пробелами, при этом не имеет значения, стоит ли один или 100 пробелов подряд.
Одна или несколько пустых строк означают конец абзаца.

```
Слова и предложения
разделяются пробелами,
при этом не имеет
значения, стоит
ли один или          100
пробелов подряд.

Одна или несколько
пустых строк означают
конец абзаца.
```

Формат абзаца определяется стилем документа. Форматы article, report и book начинают каждый абзац с красной строки — как это делается во всех других разделах нашего описания. В письмах (letter) — или если в текст включено определение

```
\setlength{\parindent}{0pt}
\setlength{\parskip}{5pt plus 2pt minus 1pt}
```

— абзацы разделяются вертикальным промежутком (как в этом разделе). Побочным эффектом такой команды является изменение промежутков в различных “окружениях” и при наборе формул.

Описанные в разделе 2.8 “окружения” (environments) позволяют набирать отдельные куски текста особым образом.

При необходимости можно явно указать требуемое разбиение на строки: команда `\\` или `\newline` означает переход на новую строку (новый абзац не начинается); команда `\\*` означает переход на новую строку, причем в этом месте не допускается переход на новую страницу. Команда `\newpage` вызывает переход на новую страницу. Команды `\linebreak[n]`, `\nolinebreak[n]`, `\pagebreak[n]` и `\nopagebreak[n]` позволяют сказать, что в этом месте переход на новую строку или страницу желателен или нежелателен, причем n указывает “силу” этого желания или нежелания (возможные значения 1, 2 или 3).

LaTeX очень старается сделать переходы со строки на строку как можно лучше. Если ему не удается удовлетворить своим (строгим) требованиям, то возникает ошибка (“overfull hbox”), и соответствующая строка остается слишком длинной. (Часто это бывает, если нет хорошего места для переноса.) С помощью команды `\sloppy` можно попросить LaTeX быть более терпимым к длинным пробелам в строке. Часто это позволяет выровнять строки, хотя строки с чрезмерно большими интервалами между словами остаются (выдается предупреждение “underfull hbox”).

2.1.2 Переносы

Если автоматический механизм переносов в конкретных ситуациях работает плохо, можно управлять переносом вручную. (Особенно это полезно для иностранных слов.)

Команда `\hyphenation` означает, что указанные слова (где бы они после этого не встречались) можно переносить только в отмеченных с помощью знака “`-`” местах. Она работает для слов, состояших только из букв без Umlaut-ов (как у буквы ё), цифр и т.п. Например,

```
\hyphenation{ Рас-свет
  Рас-светы FORTRAN }
```

Команда `\-` означает, что данное вхождение слова можно переносить только в тех местах, где стоят знаки `\-`. Она годится для любых слов (с буквой ё, цифрами и т.д.)

Рассвет, LaTeX-команда, наёмник

```
Рас\-свет,
\LaTeX-ко\-ман\-да,
на\-\"ем\-ник
```

Особый механизм для языков, где при переносе слова меняются буквы, задается командой `\discretionary`.

Команда `\mbox{...}` запрещает разрывать свой аргумент.

Прошу не звонить по телефону (0222) 5601-3694.
Параметр *filename* задает имя файла.

```
Прошу не звонить по телефону
\mbox{(0222) 5601-3694}. \\
Параметр \mbox{\it filename\/}
задает имя фай\-ла.
```

2.2 Спецзнаки

2.2.1 Кавычки

В отличие от пишущей машинки книжный набор использует различные знаки для открывающей и закрывающей кавычек (вместо нейтрального знака `"`). Открывающая кавычка изображается во входном тексте двумя подряд идущими обратными апострофами, закрывающая — двумя апострофами.

“Нет,” сказал он, “я не знаю!”

```
``Нет,'' сказал он,
``я не знаю!''
```

2.2.2 Дефисы, минусы и тире

Различные виды этих знаков изображаются одним или несколькоми знаками минуса.

X-лучи
10–18 часов
ja – oder nein?
yes—or no?
0, 1 und -1

```
X-лучи \\
10--18~часов \\
ja -- oder nein? \\
yes---or no? \\
0, 1 und $-1$
```

2.2.3 Многоточия (dots)

На пишущей машинке многоточие — это три точки подряд (каждая из которых имеет стандартную ширину буквы). При наборе это не так: для многоточия есть специальная команда `\ldots` или `\dots`.

Вместо ... пишем так:
Москва, Петушки, . . .

```
Вместо ... пишем так: \\
Москва, Петушки, \dots
```

2.2.4 Лигатуры

При наборе некоторые комбинации рядом стоящих букв набираются как один символ или сближаются:

ff fi fl AV Te. . . вместо ff fi fl AV Te. . .

Этого можно избежать, вставив между желающими слиться буквами команду `\/` или `{\kern0pt}`. Так разумно делать в том случае, если эти буквы принадлежат разным слогам.

Не Auflage (Au-fl-age),
но Auflage (Auf-lage)

```
Не Auflage (Au-fl-age),\\
но Auf\/lage (Auf-lage)
```

Таблица 3: Акценты и прочее

Набрано	Вышло	Набрано	Вышло
`\‘o`	ò	`\’o`	ó
`\^o`	ô	`\~o`	õ
`\=o`	ō	`\.o`	ȯ
`\u o`	ŏ	`\v o`	ǒ
`\H o`	ő	`\"o`	ö
`\c o`	o̧	`\d o`	ọ
`\b o`	o̲	`\t oo`	o͡o
`\oe`	œ	`\OE`	Œ
`\ae`	æ	`\AE`	Æ
`\aa`	å	`\AA`	Å
`\o`	ø	`\O`	Ø
`\l`	ł	`\L`	Ł
`\i`	ı	`\j`	ȷ
`!‘`	!‘	`?‘`	?‘
`\ss`	ß		

2.2.5 Акценты и т.п.

В LaTeX есть много видов акцентов и других символов над и под буквами (смотри таблицу 3, где для примера они изображены над буквой “o”). Если точка над “i” или “j” мешает акценту, надо ее удалить с помощью команд `\i` или `\j`. В русском шрифте буква ё набирается как `\"e`.

Hôtel, naïve, smørebrød.
!‘Señorita!

```
H\^otel, na\"\i ve,
sm\o rebr\o d. \\
!‘Se\~norita!
```

2.3 Промежутки между словами

Чтобы правый край был ровным, промежутки между словами варьируются. При этом после точки или вопросительного знака в конце предложения делается чуть больший промежуток. Применяется эвристика: точка после прописной буквы считается концом инициала (промежуток после нее не увеличивается), точка после строчной буквы (перед пробелом) считается концом предложения (промежуток увеличивается).

Можно попросить LaTeX отступить от этих правил: обратная косая черта перед пробелом означает, что этот промежуток не увеличивается, команда `~` (тильда) обозначает межсловный промежуток, который не надо увеличивать и который нельзя разрывать при переносе со строки на строку, команда `\@`, стоящая перед точкой, означает, что эта точка заканчивает предложение (даже если перед ней большая буква).

Таблица 4: Надписи

`\contentsname`	Contents	Оглавление
`\listfigurename`	List of Figures	Список иллюстраций
`\listtablename`	List of Tables	Список таблиц
`\abstractname`	Abstract	Реферат
`\refname`	References	Список литературы
`\bibname`	Bibliography	Библиография
`\indexname`	Index	Предметный указатель
`\figurename`	Figure	Рис.
`\tablename`	Table	Таблица
`\partname`	Part	Часть
`\chaptername`	Chapter	Глава
`\appendixname`	Appendix	Приложение

Газета Сов. Россия и т.п. издания.
к. ф. м. н. Подколесин.
… 5 м ширины.
Все покупайте витамин С. Поняли?

```
Газета Сов.\ Россия и
т.п.\ издания. \\
к.~ф.~м.~н.~Подколесин. \\
\dots\ 5~м ширины.\\
Все покупайте витамин~C\@.
Поняли?
```

Кроме того, можно с помощью команды

```
\frenchspacing
```

вообще отменить особый статус промежутков после концов предложений. (Это используется в не-английских текстах, и тогда команды `\ ` и `\@` не используются.)

2.4 Подписи, названия частей, даты

Обычно в LaTeX используются английские слова (“Contents” для оглавления, “Figure” для подрисуночной подписи и т.п. Это можно изменить, сказав, например,

```
\renewcommand{\contentsname}{Оглавление}
```

после чего вместо английского слова “Contents” будет использоваться русское “Оглавление”. Таблица 4 содержит список команд, подлежащих такому изменению (для разных стилей документов)

2.5 Разделы и подразделы (section)

Начало раздела или подраздела указывается с поиощью команды `\section{...}` или аналогичной. Иерархия таких команд включает:

Стиль “article”

```
\section \subsection \subsubsection
```

Стиль “report” или “book”:

```
\chapter \section \subsection \subsubsection
```

Благодаря этому статья может быть легко переделана в главу книги.

Промежутки между главами, нумерация разделов, колонтитулы — все это делается автоматически.

Заголовок статьи создается командой `\maketitle`. Предварительно должны быть указаны название (команда `\title`), автор (команда `\author`) и дата (`\date`).

Команда `\tableofcontents` печатает оглавление. Сведения о номерах страниц и т.п. для оглавления берутся из результатов предыдущего запуска системы LaTeX, поэтому при их изменении для получения корректного результата надо запустить программу дважды.

Команды со звездочкой (`\section*{...}` и ей подобные) не нумеруют разделы и не включают их в оглавление.

Примеры см. на рис. 2 на с. 9.

С помощью команд `\label` und `\ref` можно включать автоматические формируемые номера рисунков, таблиц и т.п. в текст. Команда `\ref{...}` превращается в номер того рисунка (таблицы и т.п.), которая помечена командой `\label{...}`. (Номер определяется по результатам предыдущего запуска.) Вот пример:

```
\section{Алгоритмы}
...
Доказательство смотри в разделе~\ref{bew}.
...
\section{Доказательства} \label{bew}
...
```

2.6 Сноски (footnote)

Сноски[1] автоматически нумеруются и помещаются в низ страницы. (Знак сноски — не обязательно цифра: при желании можно указать и другой вид сноски с помощью специальных команд.)

```
Сноски\footnote
{Это --- сноска}
автоматически...
```

2.7 Выделенные слова (emphasize)

Шрифтовые выделения осуществляются командой `\em` (emphasize), изменяющей шрифт до конца текущей группы (обычно до закрывающей фигурной скобки); эта команда дается *после* открывающей фигурной скобки.

Эта скобка стоит *перед*, а не *после* команды.

```
Эта скобка стоит {\em перед},
а не {\em после\/} команды.
```

Выделенный текст набирается курсивом*, если основной текст набран прямым шрифтом, и* прямым шрифтом*, если основной текст набран курсивом.* Команда `\/` увеличивает пробел между последней курсивной буквой и следующей за ней, необходимый из-за наклона.

Das *Nacht*lager
ist *nicht* brauchbar.
Das *Nacht*lager
ist *nicht* häßlich.

```
Das {\em Nacht}lager ist
{\em nicht} brauchbar. \\
Das {\em Nacht\/}lager ist
{\em nicht\/} h\"a\ss{}lich.
```

---

[1] Это — сноска

2.8 Окружения

Окружения позволяют набрать отдельные части текста специальным образом. Они имеют вид

`\begin{`*имя*`}` *текст* `\end{`*имя*`}`

и могут вкладываться друг в друга, образуя правильную скобочную структуру:

```
\begin{aa}...\begin{bb}...\end{bb}...\end{aa}
```

2.8.1 Цитаты (quote, quotation, verse)

Окружение типа `quote` предназначено для коротких цитат, примеров и т.п. Текст отодвигается от краев.

Типографское правило гласит, что

> никакая строка не должна быть длиннее 66 символов

Поэтому текст в газетах набирается в несколько колонок.

```
Типографское правило
гласит, что
\begin{quote}
никакая строка не должна
быть длиннее 66 символов
\end{quote}
Поэтому текст в газетах
набирается в несколько
колонок.
```

Окружение `quotation` предназначено для длинных цитат, состоящих из нескольких абзацев.

Окружение `verse` предназначено для стихотворений (или других случаев, когда требуется определенное разбиение на строки). Строки разделяются командой `\\`, а строфы — пустой строкой.

2.8.2 Перечни (itemize, enumerate, description)

Окружение `itemize` предназначено для простейших перечней (смотри рис. 3). Окружение `enumerate` предназначено для нумерованных перечней (смотри рис. 4). Окружение `description` предназначено для перечней, в которых каждый пункт имеет заголовок (например, словарных статей или иных описаний, смотри рис. 5).

2.8.3 flushleft, flushright, center

Окружения `flushleft` и `flushright` прижимают строки к левому (соответственно правому) краю. Окружение `center` помещает строки посередине между ними. Строки разделяются командой `\\`. (Если этого не сделать, то разбиение на строки будет сделано автоматически.)

левый
марш

```
\begin{flushleft}
левый \\
марш
\end{flushleft}
```

наше дело
правое

```
\begin{flushright}
наше дело \\
правое
\end{flushright}
```

а
мы
всегда
в центре

```
\begin{center}
а \\ мы \\
всегда \\ в центре
\end{center}
```

Перечни:

- В окружении `itemize` элементы перечня снабжаются кружочками и другими символами.
- Перечни могут быть вложенными друг в друга:
    - Максимально допустимая глубина вложенности равна 4.
    - Отступы и символы перед элементами перечня выбираются автоматически.
- и т.д.

```
Перечни:
\begin{itemize}
\item В окружении {\tt
itemize} элементы...

\item Перечни могут быть
вложенными друг в друга:
  \begin{itemize}
  \item Максимально...
  \item Отступы и...
  \end{itemize}

\item и т.д.

\end{itemize}
```

Рис. 3: Примеры окружений `itemize`

Нумерованные перечни:

1. В окружении `enumerate` элементы списка нумеруются цифрами или буквами.
2. Нумерация производится автоматически.
3. Перечни могут быть вложенными друг в друга:
    (a) Максимально допустимая глубина вложенности равна 4.
    (b) Отступы и обозначения для элементов перечня выбираются автоматически.
4. и т.д.

```
Нумерованные перечни:
\begin{enumerate}

\item В окружении {\tt
enumerate} элементы...

\item Нумерация...

\item Перечни могут быть
вложенными друг в друга:
  \begin{enumerate}
  \item Максимально...
  \item Отступы и...
  \end{enumerate}

\item и т.д.

\end{enumerate}
```

Рис. 4: Пример окружения `enumerate`

Это должен знать каждый:

- **8086:** старинный процессор, трогательно беспомощно вычисляющий адрес.
- **80286:** менее старинный процессор, трогательно скрывающий свое отличие от предыдущего.
- **80486/80586:** последний писк фирмы Intel, которую нелег- ко отклонить от выбранного пути.

```
Это должен знать каждый:
\begin{description}

\item[8086:]
старинный процессор,...

\item[80286:] менее
старинный процессор,...


\item[80486/80586:]
последний писк фир\-мы
Intel, которую...
\end{description}
```

Рис. 5: Пример окружения description

2.8.4 Буквальное воспроизведение (verbatim, verb)

Текст, окруженный строками `\begin{verbatim}` и `\end{verbatim}`, печатается в точности так, как в файле, шрифтом фиксированной ширины (`tt`), со всеми пустыми строками. Все команды остаются при этом неинтерпретированными. Это полезно, например, при наборе кусков программ.

Короткие последовательности символов (включая спецсимволы) удобно набирать (для буквального воспроизведения) между командами `\verb|` и `|` Именно так в данном руководстве набраны все команды системы LaTeX.

Команда `\dots` …

```
Команда \verb|\dots| \dots
```

Окружение `verbatim` и команда `\verb` *не* могут быть использованы внутри других команд, а также внутри окружения `tabular`.

2.8.5 Рисунки (figure)

Стоящий между `\begin{figure}` и `\end{figure}` текст — например, оставленное с помощью команды `\vspace` место для приклеивания картинки — автоматически размещается в таком месте, где он укладывается целиком (не переходя со страницы на страницу). Команда `\caption{...}` позволяет сделать подрисуночную подпись. Эта подпись состоит из слова “Figure”, порядкового номера картинки, присвоенного ей программой LaTeX, и указанной в команде подписи. Подпись размещается *под* рисунком. Команды `\label` и `\ref` позволяют пометить рисунок, а затем использовать эту метку для обозначения (неизвестного в момент набора текста) его номера.

Рис. 6 — это шедевр модернизма.

```
Рис.~\ref{weiss} --- это
шедевр модернизма.
\begin{figure}
\vspace{6cm}
\caption{Саша Черный.
  Белым-бело} \label{weiss}
\end{figure}
```

Рис. 6: Саша Черный. Белым-бело

2.8.6 Таблицы (table)

Команды для таблиц `\begin{table}` и `\end{table}` аналогичны командам для рисунков. При этом действуют и команды `\caption`, `\label`, `\ref`. Для таблиц подпись следует делать либо всегда над таблицей, либо всегда под.

Для формирования тела таблицы используется обычно окружение `tabbing` или `tabular` (см. ниже), хотя можно использовать и любое другое, например `enumerate` или `description`. Рис. 7 показывает одну из возможностей.

```
\begin{table}
\caption{...} \label{...}
  \begin{center}
    \begin{tabular}{...}
    ...
    \end{tabular}
  \end{center}
\end{table}
```

Рис. 7: Использование окружения `table`

2.8.7 Табуляторы (tabbing)

Окружение `tabbing` напоминает работу табуляторов в пишущей машинке. Команда \= устанавливает табулятор в заданном месте. Команда `\kill` означает, что данная строка — это образец, указывающий места табуляторов, и печатать ее не надо. Команда \> перескакивает к следующему табулятору, а команда \\ разделяет строки.

лево	середина	право
Мой		
дядя самых	честных	правил
он		уважать себя
		заставил

```
\begin{tabbing}
дядя самых\quad\=
середина\quad\=\kill
лево\>середина\>
право\\
Мой\\
дядя самых\>честных
 \> правил\\
он\>\>уважать себя\\
 \> \> заставил
\end{tabbing}
```

2.8.8 Собственно таблицы (tabular)

Окружение `tabular` предназначено для таблиц, в которых ширина колонок выбирается автоматически (по максимальной ширине их содержания). Имеются средства для набора отрезков, разделяющих колонки и окружающих таблицу.

Формат таблицы определяется строкой символов, являющейся параметром команды `\begin{tabular}{...}`. Именно, `l` означает колонку, в которой текст у левого края, `r` — у правого, `c` — колонку с центрированным текстом. Запись `p{`*ширина*`}` внутри параметра команды `\begin{tabular}{...}`. означает колонку заданной ширины, в которой текст занимает несколько строчек; запись `|` — вертикальный отрезок.

Внутри таблицы знак `&` означает переход к содержимому следующей колонки, комбинация знаков `\\` разделяет строки таблицы, а команда `\hline` (на месте строки таблицы) означает горизонтальный отрезок, разделяющий строки таблицы.

2201111	троичная
3700	восьмеричная
11111000000	двоичная
1984	десятичная

```
\begin{tabular}{|rl|}
\hline
2201111 & троичная \\
3700 & восьмеричная \\
11111000000 & двоичная\\
\hline
1984 & десятичная\\
\hline
\end{tabular}
```

3 Как набирать формулы

3.1 Общие замечания

Формулы внутри текста окружаются командами `\(` и `\)`, знаками `$` (с обеих сторон) или командами `\begin{math}` und `\end{math}` (годится любой из вариантов). Формулами считаются как целые формулы, так и отдельные буквы, в том числе греческие, а также верхние и нижние индексы и спецзнаки.

Катеты a,b треугольника связаны с гипотенузой c формулой $c^{2}=a^{2}+b^{2}$ (теорема Пифагора).

```
Катеты $a$,$b$ треугольника
связаны с гипотенузой $c$
формулой $c^{2}=a^{2}+b^{2}$
(теорема Пифагора).
```

TEX произносится как $\tau\epsilon\chi$.

100 m^{2} площади

Mit $\heartsuit$-lichen Grüßen

```
\TeX\ произносится как
 $\tau\epsilon\chi$.\\[6pt]
100~m$^{2}$ площади \\[6pt]
Mit $\heartsuit$-lichen
 Gr\"u\ss{}en
```

Выделяемые в отдельную строку формулы окружаются комбинациями символов `\[` и `\]` или командами `\begin{displaymath}` und `\end{displaymath}`. Если формула в отдельной строке должна быть пронумерована, ее помещают между командами `\begin{equation}` и `\end{equation}`,

Катеты a, b треугольника связаны с гипотенузой c формулой

$$c = \sqrt{a^{2}+b^{2}} \tag{1}$$

(теорема Пифагора).

```
Катеты $a$, $b$ треугольника
связаны с гипотенузой $c$
формулой
\begin{equation}
c = \sqrt{  a^{2}+b^{2}  }
\end{equation}
(теорема Пифагора).
```

С помощью команд `\label` и `\ref` номера формул могут быть указаны в тексте.

$$\epsilon > 0 \tag{2}$$

Из (2) следует …

```
\begin{equation} \label{eps}
\epsilon > 0
\end{equation}

Из (\ref{eps}) следует \dots
```

Правила набора в математическом режиме отличаются от текстовых. Именно:

1. Пробелы и переходы со строки на строку не имеют значения. Промежутки определяются автоматически по структуре формулы, но могут быть изменены явными командами типа `\,` или `\qquad`.

$$\forall x \in \mathrm{R}: \qquad x^{2} \geq 0 \tag{3}$$

```
\begin{equation}
\forall x \in
\mbox{\rm R}:
\qquad x^{2} \geq 0
\end{equation}
```

2. Пустые строки не разрешаются.

3. Каждая буква рассматривается как имя переменной и набирается курсивом с сответствующим отступом. Если внутри формулы встречается обычный текст, его надо заключить в `\mbox{...}`.

$$x^{2} \geq 0 \qquad \mbox{для всех } x \in \mathrm{R} \tag{4}$$

```
\begin{equation}
x^{2} \geq 0\qquad
\mbox{для всех }
x \in \mbox{\rm R}
\end{equation}
```

3.2 Основные элементы формул

Список спецсимволов смотри в разделе 3.5.

Маленькие **греческие буквы** набираются как `\alpha`, `\beta`, `\gamma` и т.д. Большие греческие буквы набираются как `\mbox{\rm A}`, `\mbox{\rm B}`, `\Gamma`, `\Delta`, и т.д.

$\lambda, \xi, \pi, \mu, \Phi, \Omega$

```
$\lambda, \xi, \pi, \mu,
 \Phi, \Omega $
```

Степени и индексы набираются с помощью знаков `^` и `_` соответственно.

$a_{1} \qquad x^{2} \qquad e^{-\alpha t} \qquad a^{3}_{ij}$

```
$a_{1}$ \qquad
$x^{2}$ \qquad
$e^{-\alpha t}$ \qquad
$a^{3}_{ij}$
```

Знак **корня** набирается с помощью команды `\sqrt`. Корень n-ой степени набирается с помощью команды `\sqrt[`n`]`. Размер символа корня подбирается автоматически.

$\sqrt{x} \qquad \sqrt{ x^{2}+\sqrt{y} } \qquad \sqrt[3]{2}$

```
$\sqrt{x}$ \qquad
$\sqrt{ x^{2}+\sqrt{y} }$
\qquad $\sqrt[3]{2}$
```

Команды `\overline` und `\underline` проводят **горизонтальную черту** над или под выражением.

$\overline{m+n}$

```
$\overline{m+n}$
```

Команды `\overbrace` и `\underbrace` рисуют **фигурные скобки** над или под выражением.

$\underbrace{a+\cdots+z}_{26}$

```
$\underbrace{a+\cdots+z}_{26}$
```

Штрихи, стрелки, тильды и т.п. набираются с помощью команд таблицы 5. Длинные тильды и шапочки (до 3 букв под ними) набираются с помощью `\widetilde` и `\widehat`. Штрихи набираются с помощью символа `'`.

$$y=x^{2} \qquad y'=2x \qquad y''=2$$

```
\begin{displaymath}
y=x^{2} \qquad
y'=2x   \qquad
y''=2
\end{displaymath}
```

Символы **стандартных функций** принято набирать не курсивом, а прямым шрифтом (в отличие от переменных), для чего можно использовать команды

```
\arccos \cos  \csc \exp \ker    \limsup \min \sinh
\arcsin \cosh \deg \gcd \lg     \ln     \Pr  \sup
\arctan \cot  \det \hom \lim    \log    \sec \tan
\arg    \coth \dim \inf \liminf \max    \sin \tanh
```

Для сравнений по модулю есть две команды: `\bmod` используется в выражениях типа $a \bmod b$; команда `\pmod{...}` — в выражениях типа $x \equiv a \pmod{b}$.

$$\lim_{x \to 0} \frac{\sin x}{x} = 1$$

```
\begin{displaymath}
\lim_{x \to 0}
\frac{\sin x}{x} =1
\end{displaymath}
```

Дроби (fraction) набираются как `\frac{...}{...}`. Для несложных дробей можно использовать также знак `/`.

$1\frac{1}{2}$ часа

$$\frac{x^{2}}{k+1}\qquad x^{\frac{2}{k+1}}\qquad x^{1/2}$$

```
$1\frac{1}{2}$~часов
\begin{displaymath}
\frac{ x^{2} }{ k+1 }\qquad
x^{ \frac{2}{k+1} }\qquad
x^{ 1/2 }
\end{displaymath}
```

Для **биномиальных коэффициентов** есть команда `\choose`. Команда `\atop` позволяет набрать то же самое без скобок.

$${n \choose k} \qquad {x \atop y+2}$$

```
\begin{displaymath}
{ n \choose k } \qquad
{ x\atop y+2 }
\end{displaymath}
```

Знак **интеграла** набирается командой `\int`, знак **суммы** — командой `\sum`. Верхние и нижние пределы набираются с помощью знаков `^` и `_`, как степени и индексы.

Обычно пределы интегрирования набираются справа от знака интеграла (для экономии места), но с помощью команды `\limits` можно набрать их снизу и сверху.

Для знака суммы границы набираются справа от знака, если сумма стоит внутри текста или если указана команда `\nolimits`. В остальных случаях они набираются снизу и сверху.

$$\sum_{i=1}^{n} \qquad \int_{0}^{\frac{\pi}{2}} \qquad \int\limits_{-\infty}^{\infty}$$

```
\begin{displaymath}
\sum_{i=1}^{n} \qquad
\int_{0}^{\frac{\pi}{2}}\qquad
\int\limits_{-\infty}^{\infty}
\end{displaymath}
```

Круглые и квадратные **скобки** набираются просто так, для фигурных скобок используются комбинации знаков `\{` и `\}`, для других есть специальные команды, например `\langle` (левая угловая скобка “$\langle$”).

Команда `\left` перед открывающей скобкой и команда `\right` позволяют автоматически выбрать нужный размер скобки.

$$1+\left(\frac{1}{1-x^{2}}\right)^3$$

```
\begin{displaymath}
1+\left(\frac{1}{1-x^{2}}
    \right) ^3
\end{displaymath}
```

В тех случаях, когда размер скобки желательно указать явно, используются команды `\bigl`, `\Bigl`, `\biggl` и `\Biggl` вместо `\left` и аналогично `\bigr` и т.п. вместо `\right`.

$$\Bigl((x+1)(x-1)\Bigr)^{2}$$

```
\begin{displaymath}
\Bigl((x+1)(x-1)\Bigr)^{2}
\end{displaymath}
```

Многоточия в формулах (например, в 1,2,. . . ,n) набираются с помощью команд `\ldots` и `\cdots`. Команда `\ldots` дает обычное многоточие (low), `\cdots` — три точки в середине строки (centered). Кроме того, есть команды `\vdots` для вертикального и `\ddots` для диагонального многоточий.

$$x_{1},\ldots,x_{n} \qquad x_{1}+\cdots+x_{n}$$

```
\begin{displaymath}
x_{1},\ldots,x_{n} \qquad
x_{1}+\cdots+x_{n}
\end{displaymath}
```

3.3 Установка промежутков между символами

Если промежутки между символами в формулах, выбранные автоматически, кажутся неудачными, их можно изменить явными командами. Важнейшие из них: `\,` для небольшого отступа, `\ ` (обратная косая черта и пробел) для среднего отступа, `\quad` и `\qquad` для большого и еще большего отступов, `\!` для уменьшения промежутка (отрицательный отступ).

$$F_{n} = F_{n-1} + F_{n-2} \qquad n \ge 2$$

```
\begin{displaymath}
F_{n} = F_{n-1} + F_{n-2}
 \qquad n \ge 2
\end{displaymath}
```

$$\int\!\!\!\int_{D} dx\,dy \quad \mbox{вместо} \quad \int\int_{D} dx dy$$

```
\begin{displaymath}
\int\!\!\!\int_{D} dx\,dy
\quad \mbox{вместо} \quad
\int\int_{D} dx dy
\end{displaymath}
```

3.4 Формулы над формулами

Для **матриц** используется окружение `array`, которое подобно окружению `tabular`, но действует в математическом режиме. Команда `\\` разделяет строки.

$$\mathbf{X} = \left(\begin{array}{ccc} x_{11} & x_{12} & \ldots \\ x_{21} & x_{22} & \ldots \\ \vdots & \vdots & \ddots \end{array} \right)$$

```
\begin{displaymath}
\mbox{\bf X} =
\left( \begin{array}{ccc}
x_{11} & x_{12} & \ldots \\
x_{21} & x_{22} & \ldots \\
\vdots & \vdots & \ddots
\end{array} \right)
\end{displaymath}
```

Для **формул из нескольких строк** используются окружения `eqnarray` и `eqnarray*`, заменяющие окружение `equation`. Внутри окружения `eqnarray` каждая строка получает свой собственный номер, в `eqnarray*` — как и в `displaymath` — номера *не* ставятся. Для системы уравнений, которой присваивается *один* номер, можно использовать `array` внутри окружения `equation`.

Окружения `eqnarray` и `eqnarray*` действуют как таблица из 3 колонок вида `{rcl}`, в которой средняя колонка предназначается для знака равенства или неравенства, и эти знаки помещаются один над другим. Команда `\\` разделяет строки.

$$\begin{array}{rcll} f(x) & = & \cos x & (5) \\ f'(x) & = & -\sin x & (6) \\ \int_0^x f(y)dy & = & \sin x & (7) \end{array}$$

```
\begin{eqnarray}
f(x) & = & \cos x       \\
f'(x) & = & -\sin x     \\
\int_{0}^{x} f(y)dy &
 = & \sin x
\end{eqnarray}
```

Длинные уравнения не переносятся автоматически, места переноса должны быть указаны явно. Вот два возможных варианта:

$$\begin{array}{rcll} \sin x & = & x-\frac{x^3}{3!}+\frac{x^5}{5!}- & \\ & & -\frac{x^7}{7!}+\cdots & (8) \end{array}$$

```
\begin{eqnarray}
\sin x&=&x-\frac{x^{3}}{3!}
    +\frac{x^{5}}{5!} -
          \nonumber\\
 &&-\frac{x^{7}}{7!}+\cdots
\end{eqnarray}
```

$$\begin{array}{ll} \multicolumn{2}{l}{\cos x = 1-\frac{x^2}{2!}+} \\ \qquad\qquad +\frac{x^4}{4!}-\frac{x^6}{6!}+\cdots & (9) \end{array}$$

```
\begin{eqnarray}
\lefteqn{ \cos x = 1
 -\frac{x^{2}}{2!} + }
          \nonumber\\
& & +\frac{x^{4}}{4!}
 -\frac{x^{6}}{6!}+\cdots
\end{eqnarray}
```

Команда `\nonumber` подавляет номер (для данной строки). Команда `\lefteqn` позволяет строке не подчиняться общим правилам выравнивания внутри `eqnarray`. Подробности смотри в *LaTeX-Manual* [1].

3.5 Спецзнаки

В следующих таблицах указаны все символы, входящие в стандартный комплект, используемый в математическом режиме.

Таблица 5: Штрихи, акценты и т.п.

`\hat a`	$\hat a$	`\check a`	$\check a$
`\tilde a`	$\tilde a$	`\acute a`	$\acute a$
`\grave a`	$\grave a$	`\dot a`	$\dot a$
`\ddot a`	$\ddot a$	`\breve a`	$\breve a$
`\bar a`	$\bar a$	`\vec a`	$\vec a$

4 Деликатесы

Полностью меню приведено в *LaTeX-Manual* [1]; здесь указаны лишь отдельные “фирменные блюда”.

Таблица 6: Маленькие греческие буквы

α	`\alpha`	ι	`\iota`	ϱ	`\varrho`
β	`\beta`	κ	`\kappa`	σ	`\sigma`
γ	`\gamma`	λ	`\lambda`	ς	`\varsigma`
δ	`\delta`	μ	`\mu`	τ	`\tau`
ϵ	`\epsilon`	ν	`\nu`	υ	`\upsilon`
ε	`\varepsilon`	ξ	`\xi`	ϕ	`\phi`
ζ	`\zeta`	o	`o`	φ	`\varphi`
η	`\eta`	π	`\pi`	χ	`\chi`
θ	`\theta`	ϖ	`\varpi`	ψ	`\psi`
ϑ	`\vartheta`	ρ	`\rho`	ω	`\omega`

Таблица 7: Большие греческие буквы

Γ	`\Gamma`	Ξ	`\Xi`	Φ	`\Phi`
Δ	`\Delta`	Π	`\Pi`	Ψ	`\Psi`
Θ	`\Theta`	Σ	`\Sigma`	Ω	`\Omega`
Λ	`\Lambda`	Υ	`\Upsilon`		

Таблица 8: Разные спецзнаки

$\aleph$	`\aleph`	$\prime$	`\prime`	$\forall$	`\forall`
$\hbar$	`\hbar`	$\emptyset$	`\emptyset`	$\exists$	`\exists`
$\imath$	`\imath`	∇	`\nabla`	$\neg$	`\neg`
$\jmath$	`\jmath`	$\surd$	`\surd`	$\flat$	`\flat`
ℓ	`\ell`	$\top$	`\top`	$\natural$	`\natural`
$\wp$	`\wp`	$\bot$	`\bot`	$\sharp$	`\sharp`
$\Re$	`\Re`	$\Vert$	`\\|`	$\clubsuit$	`\clubsuit`
$\Im$	`\Im`	“	`\angle`	$\diamondsuit$	`\diamondsuit`
∂	`\partial`	$\triangle$	`\triangle`	$\heartsuit$	`\heartsuit`
∞	`\infty`	$\backslash$	`\backslash`	$\spadesuit$	`\spadesuit`
$\mho$	`\mho`	Λ	`\Box`	Σ	`\Diamond`

Таблица 9: “Большие” символы операций

$\sum$	`\sum`	$\bigcap$	`\bigcap`	$\bigodot$	`\bigodot`
$\prod$	`\prod`	$\bigcup$	`\bigcup`	$\bigotimes$	`\bigotimes`
$\coprod$	`\coprod`	$\bigsqcup$	`\bigsqcup`	$\bigoplus$	`\bigoplus`
$\int$	`\int`	$\bigvee$	`\bigvee`	$\biguplus$	`\biguplus`
$\oint$	`\oint`	$\bigwedge$	`\bigwedge`		

Таблица 10: Символы бинарных операций

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

Таблица 11: Символы отношений

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

Таблица 12: Перечеркнутые символы

$\not<$	`\not<`	$\not>$	`\not>`	$\not=$	`\not=`
$\not\leq$	`\not\leq`	$\not\geq$	`\not\geq`	$\not\equiv$	`\not\equiv`
$\not\prec$	`\not\prec`	$\not\succ$	`\not\succ`	$\not\sim$	`\not\sim`
$\not\preceq$	`\not\preceq`	$\not\succeq$	`\not\succeq`	$\not\simeq$	`\not\simeq`
$\not\subset$	`\not\subset`	$\not\supset$	`\not\supset`	$\not\approx$	`\not\approx`
$\not\subseteq$	`\not\subseteq`	$\not\supseteq$	`\not\supseteq`	$\not\cong$	`\not\cong`
$\not\sqsubseteq$	`\not\sqsubseteq`	$\not\sqsupseteq$	`\not\sqsupseteq`	$\not\asymp$	`\not\asymp`

Таблица 13: Стрелки

$\leftarrow$	`\leftarrow`	$\longleftarrow$	`\longleftarrow`	$\uparrow$	`\uparrow`
$\Leftarrow$	`\Leftarrow`	$\Longleftarrow$	`\Longleftarrow`	$\Uparrow$	`\Uparrow`
$\rightarrow$	`\rightarrow`	$\longrightarrow$	`\longrightarrow`	$\downarrow$	`\downarrow`
$\Rightarrow$	`\Rightarrow`	$\Longrightarrow$	`\Longrightarrow`	$\Downarrow$	`\Downarrow`
$\leftrightarrow$	`\leftrightarrow`	$\longleftrightarrow$	`\longleft...`	$\updownarrow$	`\updownarrow`
$\Leftrightarrow$	`\Leftrightarrow`	$\Longleftrightarrow$	`\Longleft...`	$\Updownarrow$	`\Updownarrow`
$\mapsto$	`\mapsto`	$\longmapsto$	`\longmapsto`	$\nearrow$	`\nearrow`
$\hookleftarrow$	`\hookleftarrow`	$\hookrightarrow$	`\hookrightarrow`	$\searrow$	`\searrow`
$\leftharpoonup$	`\leftharpoonup`	$\rightharpoonup$	`\rightharpoonup`	$\swarrow$	`\swarrow`
$\leftharpoondown$	`\leftharpoondown`	$\rightharpoondown$	`\rightharpoondown`	$\nwarrow$	`\nwarrow`
Ω	`\rightleftharpoons`			-	`\leadsto`

Таблица 14: Левые скобки

$($	`(`	$[$	`[`	$\{$	`\{`
$[$	`\lbrack`	$\lfloor$	`\lfloor`	$\lceil$	`\lceil`
$\{$	`\lbrace`	$\langle$	`\langle`		

Таблица 15: Правые скобки

$)$	`)`	$]$	`]`	$\}$	`\}`
$]$	`\rbrack`	$\rfloor$	`\rfloor`	$\rceil$	`\rceil`
$\}$	`\rbrace`	$\rangle$	`\rangle`		

Таблица 16: Синонимы

Некоторые символы можно набрать несколькими способами.

$\neq$	`\ne` or `\neq`	`\not=`
$\leq$	`\le`	`\leq`
$\geq$	`\ge`	`\geq`
$\{$	`\{`	`\lbrace`
$\}$	`\}`	`\rbrace`
$\to$	`\to`	`\rightarrow`
$\gets$	`\gets`	`\leftarrow`
$\ni$	`\owns`	`\ni`
$\wedge$	`\land`	`\wedge`
$\vee$	`\lor`	`\vee`
$\neg$	`\lnot`	`\neg`
$\vert$	`\vert`	`|`
$\Vert$	`\Vert`	`\\|`

4.1 Гарнитуры и кегли (fonts)

Обычно LaTeX выбирает гарнитуру (вид шрифта) и кегль (размер) автоматически, исходя из логической структуры текста (для заголовков берется шрифт большого размера, для выделенных с помощью команды `\em` кусков — курсив и т.п.). В особых случаях может понадобиться указать вид и размер шрифта явно. Для этого существуют команды, описанные в таблицах 18 и 19. Их использование аналогично `\em` (область действия ограничена фигурными скобками).

Die kleinen **fetten** Römer beherrschten das ganze große *Italien.*

le $2^{\text{ème}}$ régime

```
{\small
Die kleinen {\bf fetten}
R\"omer beherrschten }
{\large das ganze gro\ss{}e
{\it Italien\/}.} \\ le
$2^{\mbox{\scriptsize\`eme}}$
r\'egime
```

Команды, меняющие размер (таблица 19), в некоторых версиях системы LaTeX устанавливают одновременно тип шрифта `\rm`. Поэтому полужирный большой шрифт требует не команд `\bf\large`, а команд `\large\bf`.

При изменения размера шрифта меняется автоматически и расстояние между строками и абзацами (в случае абзацев пустая строка должна входить в область действия команды, так что не ставьте закрывающую фигурную скобку слишком рано).

Чем *меньше* различных видов шрифта использовано в тексте, тем легче его читать и тем красивее он выглядит.

4.2 Промежутки

4.2.1 Интервал между строками

Увеличить интервал между строками можно, переопределив команду `\baselinestretch`:

для “полуторного” интервала:
`\renewcommand{\baselinestretch}{1.3}`

для “двойного” интервала:
`\renewcommand{\baselinestretch}{1.6}`

Таблица 17: Нематематические символы

Следующие символы можно использовать в текстовом режиме:

†	`\dag`	§	`\S`	©	`\copyright`
‡	`\ddag`	¶	`\P`	$	`\pounds`

Таблица 18: Гарнитуры

\rm	обычный шрифт (roman)
\bf	**(полу)жирный шрифт (boldface)**
\it	*курсив (italic)*
\sl	*наклонный шрифт (slanted)*
\sf	
\sc	“Caps and Small Caps”
\tt	`имитация пишущей машинки (typewriter)`
\boldmath	полужирный шрифт в формулах

Таблица 19: Размеры

\tiny	
\scriptsize	Очень маленький (как индексы)
\footnotesize	Маленький (как сноски)
\small	Маленький
\normalsize	Нормальный
\large	Большой
\Large	Еще больший
\LARGE	Очень большой
\huge	Совсем большой
\Huge	

Таблица 20: Единицы измерения длины

`mm`	миллиметр
`cm`	сантиметр = 10 mm
`in`	дюйм ≈ 25 mm
`pt`	point $\approx \frac{1}{72}$ in $\approx \frac{1}{3}$ mm
`em`	примерно ширина буквы M текущего шрифта
`ex`	примерно высота буквы x текущего шрифта

Таблица 21: Горизонтальные промежутки

`\,`	очень небольшой
`\enspace`	шириной в цифру
`\quad`	шириной в высоту буквы (“белый квадрат”)
`\qquad`	вдвое шире, чем `\quad`
`\hfill`	промежуток, растяжимый от 0 до ∞.

4.2.2 Дополнительные горизонтальные промежутки

Промежутки между словами и предложениями устанавливаются автоматически. Дополнительные промежутки задаются командой

`\hspace{`*длина*`}`

Если этот промежуток должен сохраняться также и в начале (или конце) строки, используется команда `\hspace*` вместо `\hspace`. Указание длины состоит из числа и названия единицы. Важнейшие единицы указаны в таблице 20.

Здесь   1.5 см
промежутка.

```
Здесь\hspace{1.5cm}1.5~см\\
промежутка.
```

Команды в таблице 21 являются сокращениями для определенных горизонтальных промежутков. Команда `\hfill` позволяет заполнить остающееся место. Примеры использования команд `\,` и `\hfill`:

“ ‘Parsifal’ — это не ‘Cats’.”

```
``\,`Parsifal' --- это не
`Cats'.''
```

левой   правой

```
левой\hfill правой
```

4.2.3 Дополнительные вертикальные промежутки

Промежутки между абзацами, главами и т.п. выбираются автоматически. При необходимости можно указать необходимый промежуток *между двумя данными абзацами* с помощью команды

`\vspace{`*длина*`}`

Таблица 22: Вертикальные промежутки

`\smallskip`	примерно $\frac{1}{4}$ строки
`\medskip`	примерно $\frac{1}{2}$ строки
`\bigskip`	примерно 1 строка
`\vfill`	промежуток, растяжимый от 0 до ∞

Эта команда должна быть окружена пустыми строками. Если промежуток не должен исчезнуть даже в начале или конце страницы, нужно использовать команду `\vspace*` вместо`\vspace`. Команды в таблице 22 представляют собой сокращения для вертикальных промежутков определенных длин. Команда `\vfill` в сочетании с `\pagebreak[4]` позволяет прижать текст к верху страницы или (будучи использована в начале и конце страницы) разместить его по центру.

Дополнительные промежутки между отдельными строками *внутри* абзаца или таблицы могут быть указаны командой

`\\[`*длина*`]`

4.3 Размер бумаги и его учет

Стандартная A4-бумага (210 × 297 *mm*) чуть выше и уже, чем используемая в США (8.5 × 11 *in*), на которую рассчитан LaTeX. С помощью команд

```
\addtolength{\topmargin}{-...}
\addtolength{\textheight}{...}
```

можно уменьшить верхнее поле и увеличить размер всех страниц. Изменение величины `\textheight` должно быть кратно величине `\baselineskip`, которая зависит от размера шрифта. Рекомендуемые значения таковы:

Размер шрифта (кегль)	10pt	11pt	12pt
`baselineskip`	12pt	13.6pt	15pt
Δ`topmargin`	−30pt	−34pt	−30pt
Δ`textheight`	60pt	68pt	60pt

Кроме того, если верхняя или нижняя служебная строка пуста, можно чуть увеличить или уменьшить добавку к `\topmargin`. Например, это руководство напечатано с помощью команд

```
\addtolength{\topmargin}{-27pt}
\addtolength{\textheight}{68pt}
```

Вообще говоря, такие изменения рекомендуется делать *не* прямо в тексте, а в файле, указывающем стиль документа, см. [3].

Длина строки и левое и правое поле не должны меняться, поскольку многие другие параметры должны быть с ними согласованы. Дополнительные сведения об изменении форматов документов смотри в [12].

4.4 Письма (letter)

Используя стиль `letter` можно поместить одно или несколько писем между командой `\begin{document}` и командой `\end{document}`.

Команды `\signature` и `\address` определяют имя и адрес отправителя. Команда `\begin{letter}{...}` начинает письмо. Ее параметром является имя и адрес адресата. Команды `\opening{...}` и `\closing{...}` задают обращение и заключительную фразу, которая автоматически помещается перед подписью, указанной командой `\signature`. Команда `\end{letter}` завершает очередное письмо.

Если перед `\begin{document}` стоит команда `\makelabels`, то получается этикетка с адресом.

Рис. 8 содержит пример письма.

```
\documentstyle[12pt]{letter}
\address{EDV-Zentrum der TU Wien \\
         Abt.~Digitalrechenanlage \\
         Wiedner Hauptstra\ss{}e 8--10 \\ A-1040 Wien }
\signature{Dr.~Hubert Partl}
\begin{document}
\begin{letter}{Frau Mag.~Elisabeth Schlegl \\
      EDV-Zentrum der Karl-Franzens-Universit\"at \\
      Attemsgasse 25/II \\ A-8010 Graz}
\opening{Liebe Frau Schlegl,}
herzlichen Dank f\"ur die Zusendung .....

..... in etwa 2--3~Wochen fertig zu sein.
\closing{Mit freundlichen Gr\"u\ss{}en}
\end{letter}
\end{document}
```

Рис. 8: Письмо H. P. к E. S.

4.5 Список литературы

Для него используется окружение `thebibliography`. В нем каждому источнику предшествует команда `\bibitem`. Ее параметром является условное название источника, которое будет указано в соответствующем месте текста с помощью команды `\cite`. Нумерация источников производится автоматически. Параметр команды `\begin{thebibliography}` указывает максимальную возможную ширину номера; например, `{99}` годится, если все номера не более чем двузначные.

Как пишет Partl [1], ...

```
Как пишет Partl~\cite{pa},
\dots
```

Список литературы

[1] H. Partl: *German TEX,* TUGboat Vol. 9, No. 1 (1988)

```
\begin{thebibliography}{99}
\bibitem{pa}
H.~Partl: {\it German \TeX,}
TUGboat Vol.~9, No.~1 (1988)
\end{thebibliography}
```

4.6 Устойчивые и хрупкие команды

Большинство команд “устойчиво”, то есть сохраняет силу во всех обстановках. Некоторые “хрупкие” команды, однако, в определенных ситуациях (внутри так называемых перемещаемых параметров) требуют защиты с помощью команды `\protect`. К числу хрупких команд относятся команды изменения размера, указанные в таблице 19, а также команда `\footnote`. Есть также (весьма редкие) случаи, когда, например, вместо `\small` нужно писать `\protect\small`. Подробно об этом смотри в *LaTeX-Manual* [1].

5 Приложение. Вокруг TEXа

Составил А. Шень

5.1 Чего не может TEX

TEX — это набор программ, предназначенный для подготовки математических текстов типографского качества. Имея текстовый (ASCII) файл с инструкциями для форматирования (к примеру, греческая буква «альфа» должна быть записана как `\alpha`) TEX подготавливает файл с указаниями о размещении текста на бумаге, который затем может быть напечатан на принтере или просмотрен на экране.

TEX не является системой типа What You See Is What You Get: наблюдая результат его работы на экране, ничего изменить нельзя (надо редактировать исходный ASCII-файл, а затем снова подвергать его обработке — и дело это, мягко говоря, не быстрое).

TEX сам знает, как нужно набирать математические формулы, и знает неплохо — но заставить его поступиться своими принципами набора формул весьма непросто (хотя теоретически и возможно). Еще труднее его использовать, если надо разместить текст на листе сложным образом (например, для рекламного листка).

TEX неудобен, если надо включить в текст картинку. С некоторым трудом можно подготовить картинку из прямоугольников с надписями, соединенных линиями, но нарисовать на полях кошку с помощью TEXа — дело гиблое. (Впрочем, есть возможность подготовить картинку с помощью Вашего любимого графического редактора и включить ее в текст.)

5.2 Апология TEXа

- TEX написан одним человеком — Дональдом Кнутом.
- TEX (исходный текст на псевдокоде, называемом WEB) полностью опубликован (в виде книги).
- TEX содержит мало ошибок: Кнут однажды сказал, что он только что исправил последнюю ошибку — и не был осмеян.
- TEX перенесен практически на все компьютеры (Unix, MS DOS, Atari, Amiga, Macintosh,...) и эти версии действительно работают одинаково.
- TEX распространяется бесплатно и свободно (не считая нескольких коммерческих версий, которые — как и должно — хуже бесплатных)
- TEX — если берется за что-то — делает это как следует: например, все делаемые им переносы в английских словах, имеющихся в словаре, делаются правильно, при наборе учитываются лигатуры и т.п.

5.3 TEX в первом приближении

Автор готовит файл с текстом, оснащенном командами для TEXа, который по традиции имеет расширение `.tex`. Далее он обрабатывает файл с помощью программы TEX и получает файл с расширением `.dvi` (device-independent — не зависящий от устройства). Этот файл является входом dvi-драйверов, которые печатают его на лазерном принтере, точечно-матричном принтере, показывают на экране и т.д. (для разных устройств есть разные драйверы). Неудовлетворенный результатом, автор вносит изменения в исходный файл — и цикл повторяется.

5.4 Шрифты

TEX может работать с разными шрифтами. При этом собственно программа TEX должна знать только размеры букв (для размещения их на странице), а внешний вид букв нужен dvi-драйверам. Поэтому эта информация хранится раздельно. Например, один из основных шрифтов cmr10 (Computer Modern Roman 10 points = кегль 10) соответствует файлам `cmr10.tfm` (размеры букв) и `cmr10.pk` (собственно буквы). При этом файл `cmr10.tfm` один для всех устройств, а файлы `cmr10.pk` для каждого разрешения принтера (количества точек на дюйм) свои. Типичное разрешение лазерного принтера — 300 точек на дюйм; LQ-принтеры в зависимости от режима работы и количества проходов дают обычно 360 × 360, 360 × 180 или 180 × 180 точек на дюйм.

Количество различных типов принтеров безгранично — поэтому появляется нужда в программе, которая могла бы генерировать шрифты в любом растре, имея в качестве исходного некоторое безрастровое описание. Такая программа входит в комплект и называется METAFONT.Исходные файлы ее обычно имеют расширение `.mf`, а результатом работы являются файлы с расширением `.tfm` (о которых уже говорилось) и файды с расширением `.gf` (это по существу растровые шрифты, но в несколько другом формате, нежели `.pk`; для их преобразования в pk-формат есть специальная программа `gftopk`).

Программа METAFONTтакже написана Кнутом и издана в виде книги (текст на псевдокоде, называемом WEB).

5.5 Макросы

По существу TEX представляет собой язык программирования — интерпретатор входных символов. Для этого интерпретатора буква «a» на входе представляет собой команду «обработай букву a» — которая интерпретируется по-разному в зависимости от внутреннего состояния интерпретатора. Среди команд есть и команды, переопределяющие значение других команд (макроопределения) — так что можно приспосабливать TEX к потребностям клиента, написав подходящий набор макроопределений. Существует несколько пакетов макроопределений — в том числе Plain TEX (написанный самим Кнутом одновременно с TEXом), LATEX (написанный Lesli Lamportom и имеющий больше возможностей), AMSTEX, AMSLATEX и другие. Практически все клиенты TEXа пользуются одним из готовых пакетов макроопределений; наиболее продвинутые добавляют к нему еще и свои макроопределения. Благодаря макроопределениям использование TEXа достаточно просто — в том случае, однако, если Вас устраивает стиль набора, навязываемый этими макроопределениями; если же нет, то Вам придется потрудиться и, возможно, написать свои собственные макроопределения.

5.6 Оптимизация: файлы формата

TEX написан довольно давно, когда и большие компьютеры были малы. Поэтому Кнут предпринял довольно много различных оптимизаций, но даже после них TEX занимает много памяти и работает медленно. Одна из наиболее заметных оптимизаций состоит в следующем: обработав часть макроопределений, TEX сбрасывает свое состояние (содержимое внутренних таблиц) в файл. Эти файлы называются файлами формата и имеют расширение `.fmt`. Например, файл формата, соответствующий пакету макроопределений `plain.tex`, называется обычно `plain.fmt`. После того как файл формата подготовлен, состояние TEXа может быть (довольно) быстро восстановлено из этого файла.

Строго говоря, механизм форматов — это не только оптимизация, так как некоторые действия (например, подготовку таблиц для переносов) можно выполнить только в режиме подготовки формата. (По крайней мере таково ограничение во многих версиях TEXа.)

Файлы формата для каждой версии TEXа свои.

5.7 METAFONT

Программа METAFONT, предназначенная для генерации шрифтов, во многом аналогична программе TEX. В ней также есть набор макроопределений `plain.mf`, после обработки которого состояние программы сливается в файл — но не с расширением `.fmt`, а с расширением `.bas` (base file). При генерации шрифта состояние восстанавливается из этого файла.

Большинство реально используемых с TEXом шрифтов (если не все) получены с помощью программы METAFONT, хотя теоретически это не обязательно. Большинство исходных текстов для METAFONTа написаны Кнутом и образуют семейство Computer Modern. В этом семействе имеются описания букв и других символов, содержащие параметры. При разных значениях этих параметров можно получить не только шрифты разного кегля (размера) — например, cmr10 есть Computer Modern Roman 10 points, а cmr5 есть Computer Modern Roman 5 points — но и разных гарнитур — например, cmss10 представляет собой рубленый (sans serif) шрифт кегля 10, и получается из того же исходного текста, что и cmr10, но при других значениях параметров. Фактически отдельные описания нужны только для строчного курсива, где формы букв принципиально отличаются от формы соответствующих прописных букв.

5.8 Разрешение и увеличение

Каждый шрифт имеет некоторый абсолютный размер (в пунктах (point)); например, абсолютный размер для шрифта cmr10 равен 10 пунктам. Шрифт может быть увеличен в некоторое число раз, называемое magnification. Стандартные увеличения образуют геометрическую прогрессию со знаменателем 1.2 (1.2 называется `\magstep 1`, 1.44 — `\magstep 2` и так далее; промежуточное увеличение ($\sqrt{1.2}$) называется `\magstephalf`. Иногда еще единичное увеличение называют «magnification 1000», так что magnification 1200 соответствует `\magstep 1`. Шрифт `cmr10 scaled\magstep1` (т.е. шрифт кегля 10, увеличенный в 1.2 раза) близок по параметрам к шрифту cmr12, имеющему кегль 12. Тем не менее требования эстетики требуют менять параметры шрифта по-разному при изменении кегля, поэтому увеличенные шрифты — это всего лишь компромисс.

Заметим, что при растрировании с разрешением 300 точек на дюйм (300 dpi — dots per inch) шрифт cmr 10 с увеличением 1200 дает в точности те же результаты, что шрифт cmr10 в натуральную величину при растрировании с разрешением 360 точек на дюйм. (А шрифт cmr12 в натуральную величину с разрешением в 300 точек на дюйм дает близкие, но не в точности такие же матрицы.) Поэтому pk-файл у них общий, и хранят его обычно в поддиректории с названием типа `360dpi`. Именно поэтому в комплекте шрифтов для одного-единственного устройства с фиксированным числом точек в дюйме можно найти шрифты разного разрешения.

5.9 TEXдля MS DOS

Существует несколько некоммерческих вариантов TEXа для MS DOS. Наиболее распространены, кажется,

- `sbTeX/sbmf` (TEX и METAFONT), авторы Wayne G. Sullivan (`WSULIVAN@IRLEARN.bitnet`) и Peter Breitenlohner [доступны для анонимного ftp в архиве `wuarchive.wustl.edu`]
- `emTeX` (также включащий METAFONT), автор Eberhard Mattes (`mattes@azu.informatik.uni-stuttgart.de`, Teckstrasse 81 D-7141 Moeglingen, West Germany) [доступны для анонимного ftp в `terminator.cc.umich.edu` (141.211.164.8), directory `msdos/text-mgmt/TeX/emtex`, а также `rusmv1.rus.uni-stuttgart.de` (129.69.1.12), directory `soft/tex/emtex`]

Считается, что sbTEX быстрее, зато emTEX может работать в меньшей памяти (он использует в качестве виртуальной памяти диск или память свыше 640 килобайт), а также имеет версию, использующую процессор 80186. Еще одна версия emTEXa использует 80386/486 в 32-разрядном режиме.

5.10 DVI-драйверы для MS DOS

Для просмотра dvi-файлов на экране существуют драйверы семейства `cdvi`, доступные по адресу CDVI SullivanSFT PO Box 292431, Lewisville, TX 75029 (написанные, видимо, автором sbTEXa). Версия 1.2 доступна бесплатно (там же, где и sbTEX), но работает только с минимальным набором шрифтов (Plain TEX), причем шрифты в нее встроены и изменены быть не могут (что делает ее бесполезной для русских букв). Зато есть варианты для всех адаптеров и работают они довольно быстро.

Другой экранный драйвер, dview, имеющийся в `wuarchive`, работает с адаптерами не хуже CGA. Его авторы — Steve Ward и Ricardo Jenez. Он может, судя по описанию, работать со шрифтами формата `*.pk` (рекомендуемое разрешение 96 точек на дюйм).

Еще один бесплатный драйвер, имеющийся в `wuarchive`, называется dvivga. (Он годится для EGA и VGA, но не для CGA.) Его написал (основываясь на более ранних разработках) Doug McDonald
(`mcdonald@cso.uiuc.bitnet`).

Наконец, в комплект emTEX также входит драйвер dviscr (однотипный с драйверами матричных принтеров dvidot и драйвером dvihplj лазерного принтера Hewlett Packard Laser Jet), который обладает массой различных возможностей, работает на всех графических адаптерах и может использовать те же шрифты, что и драйверы для принтеров. (Поскольку на экране меньше точек, чем на листе, то помещается лишь кусок страницы; можно посмотреть и большую часть страницы, регулируя «увеличение» драйвера — при этом он объединяет несколько точек шрифта в одну.)

Для матричных принтеров есть драйверы `dvieps` (8-pin Epson compatible printers), `dvipan` (24 LQ Epson-compatible, 180 × 180 dpi) и, наконец, `dvidot` из emTEXa, который умеет настраиваться на самые разные типы принтеров.

Для лазерных принтеров есть драйвер `dvihplj`, входящий в комплект emTEXa и рассчитанный на принтер Hewlett Packard Laser Jet II, а также драйвер, преобразующий текст в PostScript-формат (называется `dvips`).

Наиболее удобен, пожалуй, комплект программ emTEX, поскольку входящие в него программы согласованы друг с другом (например, если драйвер не находит нужного шрифта, то автоматически запускается программа METAFONTдля генерации этого шрифта и т.п.). Сведения об остальных программах приведены скорее для полноты картины. Поэтому, если Вы не знаете, какой вариант TEXa предпочесть, советуем начать с emTEXa. (Коммерческий вариант TEXa для MSDOS называется PC TEX, но не видно причин его использовать.)

5.11 Русские буквы в TEX

Использование TEXa для редактирования русских текстов порождает ряд проблем. Прежде всего, необходимо нарисовать русские буквы. Далее необходимо решить, в каких шрифтах эти буквы будут размещаться и по