

Содержание

2	Современные компиляторы T_EX	20
2.1	Компилятор X _Y L ^A T _E X	20
2.2	Компилятор Lua ^A T _E X	21
3	Оформление текста	22
3.1	Титульный лист	22
3.2	Заголовки, главы и разделы	24
3.3	Ввод информации в оглавление	25
3.4	Перекрестные ссылки	26
3.5	Шрифты и их размеры	27
3.5.1	Команды смены шрифта	27
3.6	Интерлиньяж	30
3.7	Разбиение на строки и страницы	31
3.7.1	Выровненные абзацы	31
3.7.2	Переносы	33
3.8	Компоновка страницы	34

2 Современные компиляторы \TeX

2.1 Компилятор \XeTeX

\XeTeX — \TeX подобная и до некоторой степени совместимая система вёрстки \XeTeX и набор классов \XeTeX входящие в основные сборки \TeX .

В отличии от классического \TeX использует кодировку UTF-8 для входных файлов, что позволяет не заботиться о выборе нужной кодировки и свободно использовать спецсимволы и символы иностранных языков доступные в Unicode. \XeTeX генерирует на выходе PDF минуя стадию DVI. Он поддерживает системные шрифты в форматах TrueType, OpenType и AAT, что позволяет использовать в документе большинство современных шрифтов. \XeTeX поддерживает добавление в документ растровых изображений в таких форматах как, например JPEG и PNG. Следует отметить, что не смотря на поддержку Unicode, такие операции, как расстановка переносов и задание типографических правил для конкретного языка выполняются отдельными пакетами: `babel` или `polyglossia`.

Рассмотрим, как изменится преамбула документа \XeTeX при подключении шрифта XITS в качестве основного, Liberation Sans в качестве шрифта без засечек и Consolas в качестве моноширного шрифта:

```

1 \documentclass[a4paper,12pt]{article}
2 \usepackage{fontspec} % подготавливает загрузку шрифтов
3 % Open Type, True Type и др.
4 \defaultfontfeatures{Ligatures={TeX}} %
5 % XITS - Коллекция научных Times-подобных шрифтов
6 % с поддержкой математического набора
7     \setmainfont{XITS}
8     \setsansfont[Scale=MatchLowercase]{Liberation Sans}
9     \setmonofont[Scale=MatchLowercase]{Consolas}
10 \usepackage{amsmath,amsfonts,amssymb,amsthm,mathtools} % AMS
11 \usepackage[math-style=french]{unicode-math}
12     \setmathfont{XITS Math}
13 \begin{document} % Конец преамбулы, начало текста.
14     {\rmfamily Пример использования шрифта в \XeLaTeX}
15     {\sffamily Пример использования шрифта в \XeLaTeX}
16     {\ttfamily Пример использования шрифта в \XeLaTeX}
17 \end{document} % Конец текста.

```

Рис. 2.1: Пример файла для \XeLaTeX с поддержкой русского языка

Следующий пример демонстрирует применение готического шрифта в \XeLaTeX —

```

{\fontspec{Deutsch Gothic.ttf}
\large Пример использования
готического шрифта}

```

**Пример использования
готического шрифта**

В дальнейшем, все примеры, приведённые в этом пособии, подготовлены для компиляции с помощью \XeLaTeX , языковым пакетом `babel` и коллекцией шрифтов XITS.

2.2 Компилятор \LuaLaTeX

\LuaLaTeX — второй современный компилятор, который является расширенной версией \PdfTeX с использованием Lua в качестве встроенного языка сценариев. В \LuaTeX основной задачей является создание открытого и настраиваемого варианта \TeX .

\LuaLaTeX также использует кодировку UTF-8, и соответственно, может использовать любые системные шрифты в форматах TrueType, OpenType и AAT.

Для компиляции документа в $\text{Lua}\text{\LaTeX}$, преамбула изменится следующим образом (подключаются те же шрифты, что и в предыдущем разделе) —

```
1 \documentclass[a4paper,12pt]{article}
2 \usepackage{fontspec} % подготавливает загрузку шрифтов
3 % Open Type, True Type и др.
4 \defaultfontfeatures{Renderer=Basic,Ligatures={TeX}} %
5 % XITS - Коллекция научных Times-подобных шрифтов
6 % с поддержкой математического набора
7     \setmainfont{XITS}
8     \setsansfont[Scale=MatchLowercase]{Liberation Sans}
9     \setmonofont[Scale=MatchLowercase]{Consolas}
10 \usepackage{amsmath,amsfonts,amssymb,amsthm,mathtools} % AMS
11 \usepackage[math-style=french]{unicode-math}
12     \setmathfont{XITS Math}
13 \begin{document} % Конец преамбулы, начало текста.
14     {\rmfamily Пример использования шрифта в \LaTeX}
15     {\sffamily Пример использования шрифта в \LaTeX}
16     {\ttfamily Пример использования шрифта в \LaTeX}
17 \end{document} % Конец текста.
```

Рис. 2.2: Пример файла для $\text{Lua}\text{\LaTeX}$ с поддержкой русского языка

3 Оформление текста

3.1 Титульный лист

Стандартный титульный лист документа в целом генерируется при помощи команды

```
\maketitle
```

Его содержимое должно быть определено командами

```
\title{...}, \author{...} и \date{...}
```

до момента вызова команды `\maketitle`. Аргумент команды `\author` может содержать несколько имен, разделенных командами `\and`.

Помимо описанных выше команд секционирования, \LaTeX 2_ε вводит три дополнительных команды для использования с классом `book`. Они полезны для деления вашей публикации. Эти команды изменяют заголовки глав и нумерацию страниц так, как это ожидается от книги:

`\frontmatter` должна быть самой первой командой после `\begin{document}`.

Она переключает нумерацию страниц на использование римских цифр. Для вводной части часто используют команды секционирования со звездочками (например, `\chapter*{Предисловие}`), чтобы \LaTeX не нумеровал эти разделы.

`\mainmatter` указывается сразу перед первой главой книги. Она включает нумерацию страниц цифрами и сбрасывает счетчик страниц.

`\appendix` отмечает начало дополнительного материала в вашей книге. После этой команды главы будут нумероваться буквами.

`\backmatter` указывается перед самыми последними частями книги, например, перед библиографией и предметным указателем. В стандартных классах документов видимого эффекта она не имеет.

При желании можно не использовать стиль оформления титульного листа, который предлагается \LaTeX ’ом. Для этого надо воспользоваться окружением `titlepage`. Текст между `\begin{titlepage}` и `\end{titlepage}` составит титульный лист, за оформление которого целиком отвечает тот, кто текст готовит. Сам \LaTeX внутри этого окружения сделает только три вещи:

- устанавливает печать в одну колонку (даже если сам документ будет печататься в две колонки);
- начинает новую страницу и устанавливает счётчик числа страниц в нуль;
- устанавливает странице стиль оформления `empty` (без колонтитула и номера).

Что и как разместить на этой странице — забота пользователя.

Также можно воспользоваться пакетом `titlepages`, который содержит более 40 вариантов титульных листов.

3.2 Заголовки, главы и разделы

Чтобы помочь читателю ориентироваться в вашей работе, вы должны разделять ее на главы, разделы и подразделы. \LaTeX поддерживает это специальными командами, принимающими в качестве аргумента заголовков раздела. Ваше дело — использовать их в надлежащем порядке.

Класс `article` включает следующие команды секционирования:

<code>\section{...}</code>	<code>\paragraph{...}</code>
<code>\subsection{...}</code>	<code>\subparagraph{...}</code>
<code>\subsubsection{...}</code>	

В классах `report` и `book` вы можете использовать дополнительную команду `\chapter{...}`.

Если вы хотите разбить ваш документ на части без изменения нумерации разделов/глав, используйте команду `\part{...}`.

Так как глав (`chapters`) в классе `article` нет, то статьи довольно легко добавлять в книгу в качестве глав. Интервалы между разделами, нумерация и размер шрифта заголовков устанавливаются \LaTeX автоматически.

Две из команд секционирования — особенные:

- команда `\part` не влияет на последовательность нумерования глав.
- команда `\appendix` аргумента не имеет. Она просто начинает нумеровать главы буквами вместо цифр.¹

\LaTeX создает оглавление, беря заголовки разделов и номера страниц из предыдущего цикла компиляции документа. Команда

`\tableofcontents`

вставляет оглавление в то место, где она вызвана. Чтобы получить правильное оглавление, новый документ должен быть обработан \LaTeX дважды. В особых случаях может быть необходим и третий проход. Когда это потребуется, \LaTeX вас предупредит.

Все вышеперечисленные команды секционирования существуют также в вариантах со звездочкой. Такой вариант получается добавлением «*» к имени команды. Они генерируют заголовки разделов, которые не нумеруются и не

¹В классе `article` меняется нумерация разделов.

включаются в оглавление. Например, команда `\section{Справка}` становится `\section*{Справка}`.

Обычно заголовки разделов появляются в оглавлении точно в том же виде, в каком они вводятся в тексте. Иногда это невозможно из-за того, что заголовков слишком длинный для оглавления. Элемент оглавления может в этом случае указываться необязательным аргументом перед собственно заголовком.

```
\chapter[Заголовок для оглавления]{Это "--- длинный,  
длинный и очень нудный заголовок, появляющийся в тексте}
```

3.3 Ввод информации в оглавление

L^AT_EX предлагает две команды для ввода информации непосредственно в файл оглавления:

```
\addtocontents{file}{text}    \addcontentsline{file}{type}{text}
```

file расширение файла оглавления, обычно `toc`, `lof` или `lot`.

type тип вхождения. Для файла с расширением `.toc` `type` обычно такой же, как для заголовков, согласно тому, в каком формате вхождение должно быть напечатано. Для файлов `.lof` или `.lot` формат задается в соответствии с `figure` или `table`.

text Информация, которая должна быть записана в `file`. Команды L^AT_EX'а следует защитить командой `\protect`, чтобы избежать нежелательных последствий.

Команда `\addtocontents` не содержит параметра `type` и предназначена для того, чтобы ввести информацию о форматировании, задаваемую пользователем. Например, если вы хотите создать дополнительный пробел в середине оглавления, нужно использовать следующую команду:

```
\addtocontents{toc}{\protect\vspace{2ex}}
```

Инструкция `\addcontentsline` обычно вызывается автоматически командами секционирования документа или командами `\caption`. Если вхождение содержит пронумерованный текст, нужно использовать команду `\numberline`, чтобы отделить номер раздела (`number`) от остального текста вхождения (`heading`) в параметре `text`:

```
\protect\numberline{number}/heading
```

Например, команда `\caption` внутри окружения `figure` сохраняет текст подрисуночной подписи следующим образом:

```
\addcontentsline{lof}{figure}%  
{\protect\numberline{\thefigure}captioned text}
```

Иногда команда `\addcontentsline` используется в документе для того, чтобы дополнить действия стандартного \LaTeX 'а. Например, в случае варианта «со звездочкой» команд секционирования в файле `.toc` не записывается никакая информация. Поэтому если вы хотите, чтобы заголовок без номера (вариант «со звездочкой») фигурировал в файле `.toc`, вы можете написать нечто вроде:

```
\chapter*{Foreword}  
\addcontentsline{toc}{chapter}{\numberline{}}Foreword}
```

Этот код создаёт вхождение на уровне «глава» с отступом в оглавлении, где вместо номера главы будет оставлен пробел. Если ликвидировать команду `\numberline`, слово «Foreword» сдвинется влево.

3.4 Перекрестные ссылки

В книгах, отчетах и статьях часто встречаются перекрестные ссылки на иллюстрации, таблицы и отдельные части текста. Для этого \LaTeX предоставляет следующие команды:

`\label{метка}, \ref{метка} и \pageref{метка}`

где *метка* — выбранный пользователем идентификатор. \LaTeX заменяет `\ref` номером раздела, подраздела, иллюстрации, таблицы или уравнения, где была использована соответствующая команда `\label`. `\pageref` печатает номер страницы, соответствующей команде `\label`.² Так же, как и в случае с заголовками разделов, здесь также используются номера из предыдущего прохода.

²Заметьте, что эти команды не знают, на что именно они ссылаются. `\label` просто сохраняет последний автоматически генерируемый номер.

Ссылка на этот

раздел~\label{sec:this}

выглядит так: <<см.

раздел~\ref{sec:this} на

стр.~\pageref{sec:this}.>>

Ссылка на этот раздел выглядит так: «см. раздел 3.4 на стр. 27.»

Для ссылок на формулы следует использовать команду `\eqref` (из пакета `amsmath`), которая отличается от команды `\ref` тем, что она автоматически ставит скобки вокруг номера формулы.

3.5 Шрифты и их размеры

3.5.1 Команды смены шрифта

\LaTeX выбирает подходящее начертание и размер шрифта, основываясь на логической структуре документа (разделы, сноски, ...). Иногда может быть желательно сменить шрифт вручную. Для этого вы можете пользоваться командами, перечисленными в таблицах 3.1 и 3.2. Действительный размер каждого шрифта определяется дизайном и зависит от класса и опций документа. Таблица 3.3 показывает абсолютные размеры, соответствующие этим командам в стандартных классах документов.

```
{\small Маленький,  
 \textbf{полужирный},  
 \Large большой,  
 \textit{курсив}.}
```

Маленький, полужирный, БОЛЬШОЙ,
курсив.

Важная особенность \LaTeX 2_ϵ заключается в том, что атрибуты шрифта независимы. Это значит, что вы можете давать команды смены размера или даже семейства шрифта, сохраняя при этом установки атрибутов наклона или насыщенности.

В *математическом режиме* вы можете использовать команды смены шрифта, чтобы временно выйти из *математического режима* и ввести нормальный текст. Если вы хотите переключиться на другой шрифт для верстки математики, то для этого существует отдельный набор команд. Смотрите таблицу 3.4.

В связи с командами смены размера шрифта заметную роль играют фигурные скобки. Они используются для построения *групп*. Группы ограничивают область действия большинства команд \LaTeX .

Таблица 3.1: Шрифты

<code>\textrm{...}</code>	прямой шрифт	<code>\textsf{...}</code>	без засечек
<code>\texttt{...}</code>	пишущая машинка		
<code>\textmd{...}</code>	нормальный	<code>\textbf{...}</code>	полужирный
<code>\textup{...}</code>	прямой шрифт	<code>\textit{...}</code>	<i>курсив</i>
<code>\textsl{...}</code>	<i>наклонный шрифт</i>	<code>\textsc{...}</code>	капитель
<code>\emph{...}</code>	<i>выделенный шрифт</i>	<code>\textnormal{...}</code>	обычный

Таблица 3.2: Размеры шрифта

<code>\tiny</code>	крошечный	<code>\Large</code>	еще больше
<code>\scriptsize</code>	очень маленький	<code>\LARGE</code>	очень большой
<code>\footnotesize</code>	довольно маленький	<code>\huge</code>	огромный
<code>\small</code>	маленький	<code>\Huge</code>	громадный
<code>\normalsize</code>	нормальный		
<code>\large</code>	большой		

Таблица 3.3: Абсолютные размеры шрифтов в стандартных классах

Размер	10pt (по умолчанию)	опция 11pt	опция 12pt
<code>\tiny</code>	5pt	6pt	6pt
<code>\scriptsize</code>	7pt	8pt	8pt
<code>\footnotesize</code>	8pt	9pt	10pt
<code>\small</code>	9pt	10pt	11pt
<code>\normalsize</code>	10pt	11pt	12pt
<code>\large</code>	12pt	12pt	14pt
<code>\Large</code>	14pt	14pt	17pt
<code>\LARGE</code>	17pt	17pt	20pt
<code>\huge</code>	20pt	20pt	25pt
<code>\Huge</code>	25pt	25pt	25pt

Таблица 3.4: Математические шрифты

Команда	Пример	Вывод
<code>\mathcal{...}</code>	<code>\$\mathcal{B}=c\$</code>	$\mathcal{B} = c$
<code>\mathrm{...}</code>	<code>\$\mathrm{K}_2\$</code>	K_2
<code>\mathbf{...}</code>	<code>\$\sum x=\mathbf{v}\$</code>	$\sum x = \mathbf{v}$
<code>\mathsf{...}</code>	<code>\$\mathsf{G\times R}\$</code>	$G \times R$
<code>\mathtt{...}</code>	<code>\$\mathtt{L}(b,c)\$</code>	$L(b,c)$
<code>\mathnormal{...}</code>	<code>\$\mathnormal{R_{19}}\neq R_{19}\$</code>	$R_{19} \neq R_{19}$
<code>\mathit{...}</code>	<code>\$\mathit{ffi}\neq ffi\$</code>	$\mathit{ffi} \neq ffi$

Ему нравятся `{\LARGE}`
большие и `{\small}`
маленькие} буквы}.

Ему нравятся **БОЛЬШИЕ И** малень-
кие **БУКВЫ.**

Команды, влияющие на размер шрифта, влияют также на расстояние между строками, но только если соответствующий абзац заканчивается внутри области действия команды. Поэтому закрывающая фигурная скобка `}` не должна стоять слишком рано. Заметьте положение команды `\par` в следующих двух примерах³:

`{\Large Не читайте это! Это
неправда. Верьте мне!\par}`

Не читайте это! Это
неправда. Верьте мне!

`{\Large Это тоже неправда.
Но помните, что я вру.}\par`

Это тоже неправда. Но
помните, что я вру.

Если вы хотите применить команду изменения размера к целому абзацу текста или больше того, то для этого лучше использовать синтаксис окружения.

`\begin{Large}`
Это неправда. Но
что в наши дни`\ldots`
`\end{Large}`

Это неправда. Но что в на-
ши дни...

Это избавит вас от подсчёта множества фигурных скобок.

³`\par` эквивалентен пустой строке.

3.6 Интерлиньяж

В полиграфии этим красивым словом называется интервал между строками. Команды наподобие `\small`, устанавливающие размер шрифта, автоматически устанавливают и размер интервала между строками, так что вручную менять его не следует (потому мы и не рассказываем, как это делать; любопытствующий читатель может узнать некоторые подробности в приложении В и все подробности в книге [2]). Можно, однако (и иногда это бывает необходимо), пропорционально увеличить или уменьшить все интервалы между строками — например, чтобы подогнать число полос в документе к требуемому. Если, скажем, вы хотите увеличить интервалы между строками на 1 %, т. е. в 1.01 раза, то в преамбуле следует написать так:

```
1 \renewcommand{\baselinestretch}{1.01}
```

Если нужно пропорционально увеличить или уменьшить интерлиньяж в каком-то фрагменте текста, то можно написать так:

```
1 {% Открывающая фигурная скобка необходима
2 \renewcommand{\baselinestretch}{1.01}
3 \selectfont
4 Текст, в котором надо изменить интерлиньяж...
5 ....
6 Конец этого текста
7 }% Закрывающая скобка, парная к открывающей
```

Также можно воспользоваться пакетом `setspace` для указания межстрочного интервала для всего документа или отдельного блока.

Если нужно изменить межстрочного интервала для всего документа, необходимо подключить пакет и добавить в преамбулу соответствующую команду —

```
1 \usepackage{setspace}
2 % для одинарного интервала
3 \singlespacing
4 % для полуторного интервала
5 \onehalfspacing
```

```

6 % двойной интервал
7 \doublespacing
8 % для произвольного интервала
9 \setstretch{line_spacing_value}

```

Для отдельного блока с текстом можно применит следующие окружения

```

1 \begin{singlespacing}
2 ...
3 \end{singlespacing}

```

или

```

1 \begin{spacing}{1.0}
2 ...
3 \end{spacing}

```

3.7 Разбиение на строки и страницы

3.7.1 Выровненные абзацы

Книги часто верстаются так, чтобы все строчки имели одинаковую длину. \LaTeX вставляет необходимые разрывы строк и пробелы между словами, оптимизируя форматирование абзаца как целого. При необходимости он также переносит слова, которые не помещаются на строке. От класса документа зависит то, как верстаются абзацы. Обычно каждый абзац начинается с красной строки, а дополнительного интервала между ними нет.

В некоторых случаях может быть необходимо указать \LaTeX разорвать

строку:

`\\` или `\newline`

начинают новую строку, не начиная нового абзаца.

`*`

запрещает, кроме того, разрыв страницы после вставленного разрыва строки.

`\newpage`

начинает новую страницу.

`\linebreak[n]`, `\nolinebreak[n]`, `\pagebreak[n]` и `\nopagebreak[n]`

разрывают строку, запрещают разрыв строки, разрывают страницу и запрещают разрыв страницы, соответственно. Необязательный аргумент n позволяет автору влиять на их действие. Он может быть равен числу от нуля до четырех. Устанавливая n в значение, меньшее 4, вы оставляете \LaTeX возможность игнорировать вашу команду, если результат будет очень плохой. Не путайте эти команды «разрыва (`break`)» с командами «начала (`new`)». Даже если вы зададите команду «разрыва», \LaTeX попытается выровнять правую границу страницы и общую высоту страницы. Если вы действительно хотите начать «новую строчку», используйте соответствующую команду.

\LaTeX всегда пытается производить наилучшее из возможных разбиений строк. Если он не может найти способ разбить строки в соответствии со своими стандартами, он позволяет одной строке выступать из абзаца вправо. \LaTeX затем выводит диагностику («`overfull hbox`») во время обработки входного файла. Чаще всего это случается, когда \LaTeX не может найти место для переноса слова.⁴ Давая команду `\sloppy`, вы можете сказать, чтобы \LaTeX несколько ослабил свои стандарты. Тогда он сможет предотвратить такие слишком длинные строки, увеличивая интервалы между словами — даже если конечный вывод будет не оптимален. В этом случае пользователь получит предупреждение («`underfull hbox`»). В большинстве случаев результат выглядит не очень хорошо. Команда `\fussy` действует в обратную сторону.

⁴Хотя \LaTeX и дает вам предупреждение, такие строки не всегда легко найти. Если вы в команде `\documentclass` используете опцию `draft`, такие строки будут отмечены толстой черной линией на правых полях.

Если использовать команду `\sloppy` внутри группы, необходимо, чтобы закрывающая фигурная скобка шла после пустой строки, завершающей абзац. Вот пример того, как нужно действовать в этом случае

```
Черепаховый суп~"--- изысканное  
деликатесное и диетическое блюдо  
{\sloppy  
  
}
```

Черепаховый суп — изысканное деликатесное и диетическое блюдо

Для более тонкого управления выбором между разреженными строками и «overflow'ами» используется параметр `\emergencystretch`. Если установить его значение равным примерно 20–30 пунктам, т. е. написать, например,

<code>\emergencystretch=25pt</code>

то в случае, когда без переполнений сверстать абзац не удастся, \TeX попыбует сделать все строки абзаца более разреженными (тем более разреженными, чем больше величина этого параметра). Точную величину `\emergencystretch` надо подбирать экспериментально.

3.7.2 Переносы

\LaTeX переносит слова, когда это необходимо. Если алгоритм переносов не находит правильных точек переноса, вы можете исправить положение, сказав \TeX об исключении при помощи следующих команд.

Команда

<code>\hyphenation{список слов}</code>
--

вызывает перенос слов, перечисленных в ее аргументе, только в местах, отмеченных «-». Эта команда должна даваться в преамбуле входного файла, и должна содержать только слова, состоящие из обычных букв. Правила переноса запоминаются для языка, активного в момент обратки команды `\hyphenation`. Это значит, что, если вы поместите ее в преамбулу документа, она будет влиять на английские переносы. Если вы поместите ее после команды `\begin{document}`, используя при этом поддержку национальных языков, например, `babel`, то переносы будут запомнены для языка, активизированного при помощи `babel`.

Следующий пример разрешает переносы в слове «hyphenation», так же, как и в слове «Hyphenation», и запрещает переносы в словах «FORTRAN», «Fortran» и «fortran». В аргументе не допускаются специальные символы.

Пример:

```
\hyphenation{FORTRAN Hy-phen-a-tion}
```

Команда `\-` вставляет в слово выборочный перенос. Он также становится единственной разрешенной точкой переноса в этом слове. Эта команда в особенности полезна для слов, содержащих специальные символы (например, символы с акцентами), потому что \LaTeX не переносит такие слова автоматически.

```
I think this is: su\~per\~cal\~%  
i\~l\~lis\~tic\~ex\~pi\~%  
al\~i\~do\~cious
```

I think this is: supercalifragilisticexpialidocious

Несколько слов можно удерживать вместе на одной строке командой

`\mbox{текст}`

Она будет в любом случае сохранять свой аргумент вместе.

Номер моего телефона скоро сменится.
Он будет `\mbox{0116 291 2319}`.

Номер моего телефона скоро сменится. Он будет 0116 291 2319.

Параметр
`\mbox{\emph{имя файла}}` должен
содержать имя файла.

Параметр *имя файла* должен содержать имя файла.

Команда `\fbox` аналогична команде `\mbox`, но вокруг ее содержимого будет нарисована видимая рамка.

3.8 Компоновка страницы

Класс документа предопределяет значения таких параметров, как ширина и высота страницы, размеры полей и пр. (с учетом опции, указывающей формат бумаги). В настоящем разделе рассказано, как изменить эти значения, если они вас не устраивают.

Размеры текста на странице, полей и пр. задаются параметрами со значением длины. Изменять эти параметры следует в преамбуле документа.

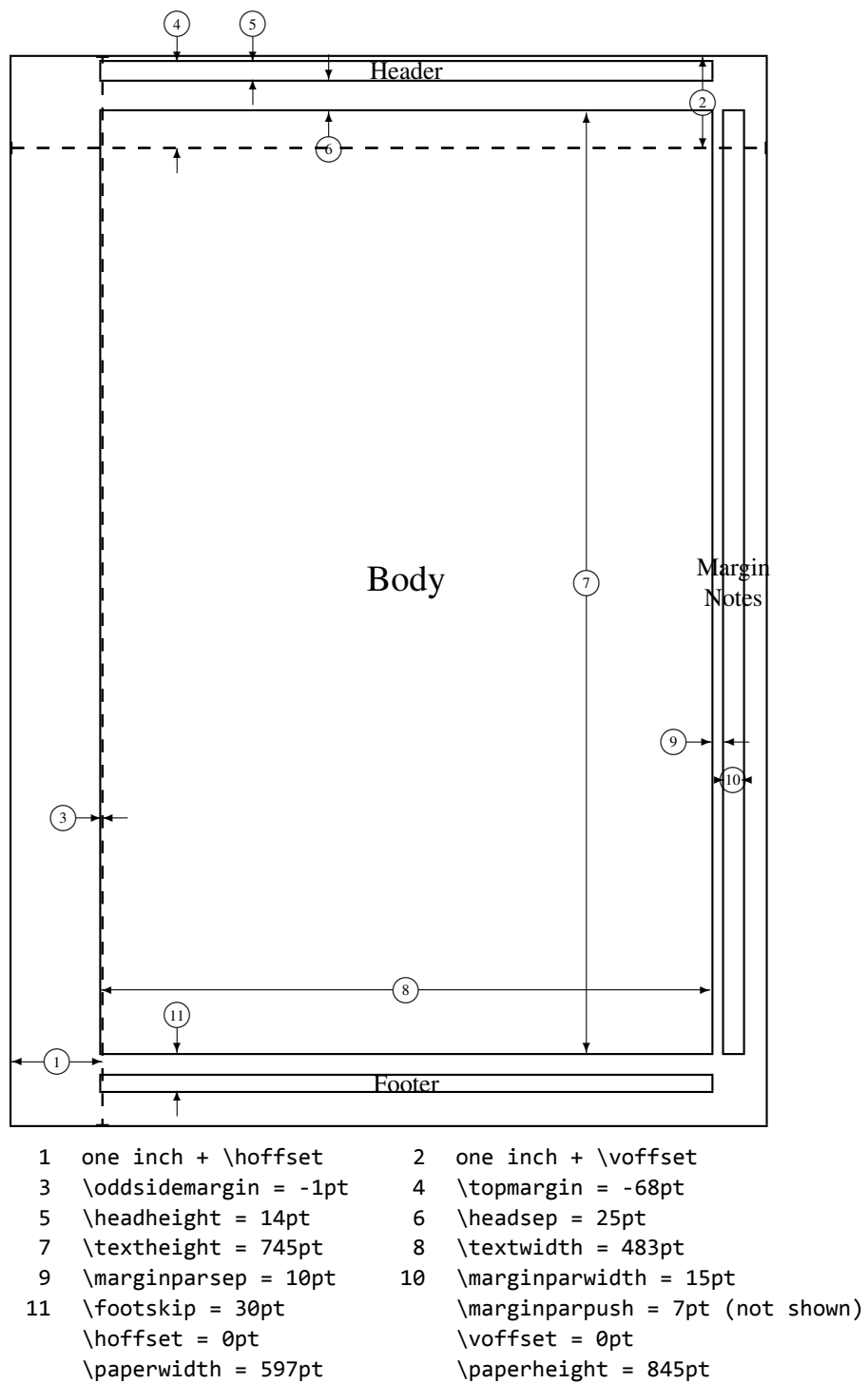


Рис. 3.1: Параметры компоновки страницы

Рисунок 3.1 показывает все параметры компоновки страницы, которые можно изменить. Однако, изменять параметры компоновки страницы стандартными средствами \LaTeX не совсем удобно. Удобный интерфейс для установки параметров страницы предоставляет пакет `geometry`. Например, чтобы изменить поля страницы, нужно в преамбулу добавить следующие строчки

```
\usepackage{geometry}  
\geometry{top=25mm} % верхнее поле  
\geometry{bottom=35mm} % нижнее поле  
\geometry{left=35mm} % левое поле  
\geometry{right=20mm} % правое поле
```

Для более подробного описания см. документацию к пакету `geometry`.