

# L<sup>A</sup>T<sub>E</sub>X, GNU/Linux и русский стиль.

© Е.М. Балдин\*



Эта статья была опубликована в февральском номере русскоязычного журнала Linux Format (<http://www.linuxformat.ru>) за 2007 год. Статья размещена с разрешения редакции журнала на сайте <http://www.inp.nsk.su/~baldin/> и до конца июля месяца все вопросы с размещением статьи в других местах следует решать с редакцией Linux Format. Затем все права на текст возвращаются ко мне.

Текст, представленный здесь, не является точной копией статьи в журнале. Текущий текст в отличии от журнального варианта корректор не просматривал. Все вопросы по содержанию, а так же замечания и предложения следует задавать мне по электронной почте <mailto:E.M.Baldin@inp.nsk.su>.

Текст на текущий момент является просто *текстом*, а не книгой. Поэтому результирующая доводка в целях улучшения восприятия текста не проводилась.

---

\*e-mail: [E.M.Baldin@inp.nsk.su](mailto:E.M.Baldin@inp.nsk.su)

Эмблемы T<sub>E</sub>X и METAFont, созданные Дуайном Бибби, взяты со странички Д.Э. Кнута. Цветной пингвин взят из пакета ps2pdf от Ральфа Найпрашека (Rolf Niepraschk)

# Оглавление

<b>6. Вёрстка I</b>	<b>1</b>
6.1. Определённые «размеры» и переменные «длины» . . . . .	1
6.2. Скелет страницы . . . . .	4
6.3. Меняем макет . . . . .	7
6.3.1. Двигаем размеры . . . . .	7
6.3.2. Стили страницы . . . . .	9
6.4. Причёсываем текст . . . . .	10
6.5. Послесловие . . . . .	13

# Глава 6

## Вёрстка I

Хороший набор — это плотный набор, «дырявый» же набор плохо читается, так как дыры нарушают связанность строки и тем самым затрудняют восприятие мысли.

---

Ян Чихольд.

Волшебных текстовых процессоров не существует. Телепатией программы пока не обладают. Они, естественно, делают то, что им сказано, но вкус и чувство прекрасного у них полностью отсутствуют. В конце занимательного приключения по созданию текстов частенько приходится брать управление в свои руки, дабы навести лоск на почти готовое произведение.

Вёрстка — составление страниц (полос) газеты, журнала, книги определённого размера из набранных строк, заголовков, иллюстраций и т. п. в соответствии с разметкой или макетом. В этой главе разберёмся с тем как задавать размеры, что такое макет полосы набора и как «удерживать» текст в рамках дозволенного.

### 6.1. Определённые «размеры» и переменные «длины»

Л<sup>A</sup>T<sub>E</sub>X поддерживает переменные типа «длина» для определения расстояния. Например, ранее уже упоминалась команда `\textwidth` — это переменная, хранящее значение длины, равной ширине текста.

Для создания переменной типа длина следует воспользоваться командой `\newlength`. В качестве обязательного параметра передаётся имя переменной. При создании переменной присваивается нулевая длина, так что следующим шагом необходимо приравнять её чему-то:

```
\newlength{\MyLen}
\setlength{\MyLen}{1cm plus 2.5fill minus 5mm}
\addtolength{\MyLen}{5em}
Длина \linline!\MyLen! равна \the\MyLen.
```

Длина `\MyLen` равна 82.89214pt  
plus 2.5fill minus 14.22636pt.

Длина в  $\LaTeX$  это не просто какой-то определённый размер — это более сложная структура с указанием границ возможного сжатия и растяжения. Границы растяжения определяются с помощью инструкции `plus`, а сжатия — `minus`. При формировании абзацев  $\TeX$  использует эту информацию для максимально «красивого» заполнения.

Команда `\setlength` эквивалентна оператору присваивания. В свою очередь команда `\addtolength` позволяет увеличить переменную на указанную величину, которая может быть отрицательной. Макрос `\the` позволяет «развернуть» переменную длины для вывода на печать.

$\LaTeX$  «говорит» в терминах англо-американской системы мер. Эта система отживает своё, но её наследие будет ещё долго проявляться и портить жизнь современному «метрическому» миру. Для определённости следует знать, что один дюйм (in) равен 2.54 сантиметра и в нём умещается 72.27 пунктов ( $1\text{ pt} \simeq 0.35\text{ mm}$ ). Метрические величины представлены привычными сантиметрами (cm) и миллиметрами (mm). Кроме упомянутых  $\LaTeX$  умеет оперировать с размерами в больших пунктах (bp), пунктах Дидо (dd), пиках (pc) и цецера (cc) — традиционные единицы измерения, используемые в типографиях. Минимальной ненулевой единицей длины в  $\LaTeX$  является приведённый пункт (sp), который составляет  $1/65536$  от одного пункта.

Кроме определённых единиц измерения длины можно задавать так же и в относительных: `1ex` соответствует высоте строчной латинской буквы `x`, а `1em` — ширине прописной латинской буквы `M`. Эти величины меняются вместе со сменой шрифта, что позволяет задавать автоматически масштабирующиеся горизонтальные промежутки не привязанные к конкретному размеру и типу шрифта. Например, широкий пробел, задаваемый с помощью команды `\quad`, определяется как `\hspace{1em}`.

```
\setlength{\MyLen}{1ex}
Высота x равна \the\MyLen\par
\Large \setlength{\MyLen}{1ex}
Высота x равна \the\MyLen
```

Высота `x` равна 4.71341pt

Высота `x` равна 7.43707pt

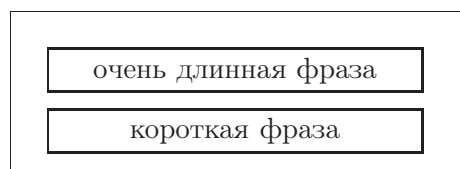
Интересной инструкцией является длина `fill` — это *бесконечность*.  $\TeX$  поддерживает операции с бесконечностями, причём оперирует тремя видами бесконечностей: `fil`, `fill` и `filll`, где `fil`  $\ll$  `fill`  $\ll$  `filll`. С помощью этих сущностей производится центрирование боксов и более сложные выравнивания.

Если хочется узнать ширину текста, то можно воспользоваться командой `\settowidth:`

```

\settowidth{\MyLen}{очень длинная фраза}
\addtolength{\MyLen}{1em}
\centering
\framebox[1.2\MyLen]{очень длинная фраза}\par
\framebox[1.2\MyLen]{короткая фраза}

```



Аналогично команда `\settoheight` позволяет выяснить высоту текста над базовой линией, а `\settodepth` — глубину под базовой линией. При использовании длины можно добавить перед ней множитель.

А теперь немного «магии» из английского FAQ по  $\text{\LaTeX}$ :

---

```

\makeatletter
\newcommand{\maxwidth}{%
  \ifdim \Gin@nat@width > \linewidth
    \linewidth
  \else
    \Gin@nat@width
  \fi
}
\makeatother

```

---

Эта конструкция определяет переменную длины `\maxwidth` таким образом, что при вставке картинки:

```
\includegraphics [width=\maxwidth] {«картинка»}
```

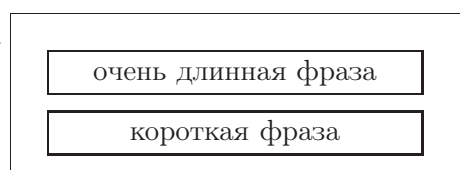
ширина картинки становится равной минимальной из двух возможных значений: «естественной» ширине картинки (размер в `BoundingBox`) или ширине строки. Это позволяет вывести картинку в натуральную величину при условии, что она не вылезает за рамки дозволенного и записывать её в эти рамки, коли она за них вылезает.

**calc** В дополнение к стандартным возможностям пакет **calc** расширяет базовые операции с длинами. Фактически **calc** вводит арифметические операции в привычной со школы инфиксной записи.

```

\setlength{\MyLen}{
  (1em+\widthof{очень длинная фраза})*\real{1.2}}
\centering
\framebox[\MyLen]{очень длинная фраза}\par
\framebox[\MyLen]{короткая фраза}

```



При загрузке **calc** `\setlength` и `\addtolength` переопределяются так, что в качестве аргумента после этого можно передавать арифметические выражения. Кроме арифметики в **calc** определяются макросы `\widthof{текст}`, `\heightof{текст}` и `\depthof{текст}` — ширина, высота и глубина текста.



Рис. 6.1. Определение ширины (width), высоты (height) и глубины (depth).

При умножении длины на число длина должна стоять до числа ( $4\text{mm} * 2$  — верно, а  $2 * 4\text{mm}$  — не верно). Делить и умножать можно только на целые числа. Действительные числа вводятся с помощью уже использованного в примере макроса `\real` и отношения длин, вычисляемого с помощью команды:

---

```
\ratio {«длина»}{«длина»}
```

---

Подробное описание пакета можно найти в документации `calc.pdf` из коллекции `tools`.

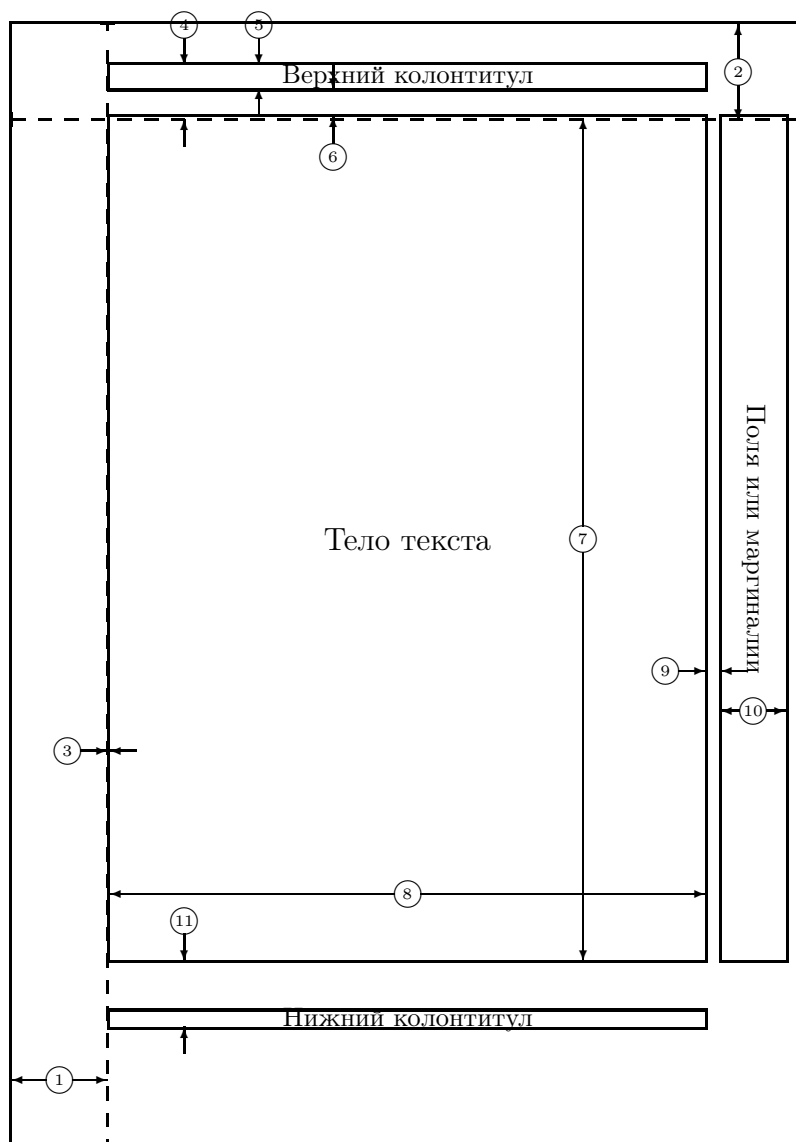
## 6.2. Скелет страницы

На рис. 6.2 приведён результат выполнения команды `\layout` из одноимённого пакета. Основное место на странице занимает текст — тело текста. Справа и слева от текста расположены поля. Поля, обычно, остаются пустыми, но иногда они используются для заметок (маргиналий или фонариков). В верхней и нижней части страницы расположены, соответственно, верхний и нижний колонтитулы. Колонтитул представляет из себя справочную строку, помогающую ориентироваться в структуре текста.

Совокупность размеров и расположений указанных полей, а так же вид и содержание колонтитулов называется *макетом полосы набора*. На рисунке пунктирной линией изображены *поля драйвера* (1 и 2) относительно которых выстраиваются все остальные поля. По договорённости отступы до полей драйвера равны одному дюйму. Переопределив `\hoffset` и `\voffset` (по умолчанию они равны нулю), можно легко сдвинуть полосу набора целиком по горизонтали и вертикали, соответственно.

Ниже перечислены параметры, которые управляют макетом полосой набора:

- Тело текста характеризуется высотой `\textheight` (7) и шириной `\textwidth` (8). При многоколоночной вёрстке ширина колонки равна `\columnwidth`. Переменная `\linewidth` принимает значение равное длине строки текущего текста.
- `\oddsidemargin` (3) добавляется слева в случае односторонней печати. При двухсторонней печати полосы набора для чётных и нечётных страниц различаются. В этом случае для нечётных слева опять же добавляется `\oddsidemargin`, а для чётных `\evensidemargin`.



1	один дюйм + <code>\hoffset</code>	2	один дюйм + <code>\voffset</code>
3	<code>\oddsidemargin = 2pt</code>	4	<code>\topmargin = -41pt</code>
5	<code>\headheight = 18pt</code>	6	<code>\headsep = 21pt</code>
7	<code>\textheight = 635pt</code>	8	<code>\textwidth = 448pt</code>
9	<code>\marginparsep = 12pt</code>	10	<code>\marginparwidth = 49pt</code>
11	<code>\footskip = 50pt</code>		<code>\marginparpush = 6pt</code> (not shown)
	<code>\hoffset = 0pt</code>		<code>\voffset = 0pt</code>
	<code>\paperwidth = 597pt</code>		<code>\paperheight = 845pt</code>

Рис. 6.2. Макет полосы набора класса `scrartcl` (опция `a4paper`). Результат выполнения команды `\layout` из пакета `layout`.

- Верхний колонтитул располагается на расстоянии `\topmargin` (4) от поля драйвера, имеет высоту `\headheight` (5), а тело текста отступает от колонтитула на расстояние `\headsep` (6).
- `\footskip` позиционирует базовую линию нижнего колонтитула относительно последней строки текста.
- Поля для заметок имеют ширину `\marginparwidth` (10) и отступают от тела текста на расстояние `\marginparsep` (9). Ещё одна опция управляет минимальным расстоянием между заметками: `\marginparpush`.

**Выбор размера бумаги** Физический размер бумаги описывается параметрами `\paperwidth` и `\paperheight`. Стандартные базовые классы L<sup>A</sup>T<sub>E</sub>X (`article`, `book`, `report` и `letter`) по умолчанию предполагают, что для печати используется бумага формата `letter`. Очевидно это умолчание не годится для России, где стандартом является формат A4 (210×297 мм). Обычно, установить правильный формат можно с помощью передачи параметра `a4paper` при выборе класса документа:

---

```
\documentclass [ a4paper , 12 pt , oneside ] { scrbook }
```

---

Для создания небольших брошюрок со страницей размера A5 (половина размера A4) используется опция `a5paper`.

**Ориентация** Для портретной и альбомной ориентации вообще-то требуют разные макеты полосы набора. Альбомная ориентация может использоваться по умолчанию в некоторых специализированных классах (например, `slides`).

Diagram illustrating the effect of `\lscapewidth`. The text "текст" is oriented vertically, rotated 90 degrees counter-clockwise. The text is arranged in a grid where each row contains 10 vertical lines of text, and there are 10 such rows. The text is rotated 90 degrees counter-clockwise.

Рис. 6.3. `\lscapewidth` в действии.

Diagram illustrating the effect of `\rotpages`. The text "задача" is oriented horizontally. The text is arranged in a grid where each row contains 10 horizontal lines of text, and there are 10 such rows. The text is rotated 90 degrees clockwise.

Diagram illustrating the effect of `\rotpages`. The text "задача" is oriented vertically, rotated 90 degrees clockwise. The text is arranged in a grid where each row contains 10 vertical lines of text, and there are 10 such rows. The text is rotated 90 degrees clockwise.

Рис. 6.4. `\rotpages` в действии.

Если повернуть требуется только текст без изменения полей колонтитулов, то проще всего воспользоваться окружением `landscape` из пакета `lscapewidth`. Всё, что находится внутри этого окружения поворачивается на 90 градусов против часовой стрелки. При использовании `pdflatex` для изменения ориентации не только текста,



но и страницы (дабы не наклонять при чтении текста голову на бок) при загрузке пакета следует передать ему опцию `pdftex` или воспользоваться пакетом-надстройкой `pdfscape`.

Лучше не менять параметры макета во время набора, но если очень хочется, то можно воспользоваться стандартным пакетом `portland`. `portland` позволяет на лету менять портретную ориентацию на альбомную и обратно, переопределяя соответствующие управляющие макетом переменные. Команды `\portrait` и `\landscape` работают как переключатели между этими режимами. В пакете определены и одноимённые с переключателями окружения. Это довольно низкоуровневые пакет и пользоваться им надо аккуратно.

Если по какой-то причине потребуется развернуть текст на 180 градусов<sup>1</sup> можно воспользоваться пакетом `rotpages`. В пакете определены два переключателя режима `\rotboxpages` и `\endrotboxpages`, которые указывают границы повернутого текста. Пакет умеет разворачивать не только страницы, но и колонки. Подробности в документации `rotpages-doc.pdf`

## 6.3. Меняем макет

Причина по которой не искушённый Т<sub>Е</sub>Xник начинает менять макет полосы набора, обычно заключается в том, что он использует стандартные классы Л<sub>А</sub>T<sub>Е</sub>X—один из четвёрки `article`, `book`, `report` или `letter`. Это очень древние классы и на них много чего «завязано», поэтому они прибывают в абсолютно замороженном состоянии. Поэтому лучше для начала найти себе класс по душе. Например, `scrartcl`, `scrbook`, `scrreprt` и `scrletter` — классы из коллекции КОМА-Script, которые дублируя функциональность стандартной четвёрки, ориентируются на европейских пользователей и размер листа А4. А.И.Роженко в рамках коллекции `ncclatex` (N<sub>C</sub>C) предоставляет класс `ncc`, ориентированный на русских Т<sub>Е</sub>Xников.

### 6.3.1. Двигаем размеры

Выбор, естественно, не ограничивается упомянутыми выше классами — он огромен. Поэтому, прежде чем что-то изменять следует внимательно изучить уже имеющиеся решения. И даже если Вы знаете что делаете, лучше не переопределять в ручную переменные управляющие размерами макета полосы набора. Правильным решением будет воспользоваться одним из уже имеющих специализированных пакетов, например: `geometry` или `vmargin`.

• Макет лучше не менять.

`geometry` Пакет `geometry` меняет размеры прямо в процессе загрузки стилевого файла, например так:

---

```
\usepackage [ height=25cm, a4paper , hmargin={3cm, 2cm} ] { geometry }
```

---

<sup>1</sup>Например, для написания задачника с решениями, чтобы усложнить обучающемуся чтение решения сразу после прочтения задачи.

Разбор параметров выполняется с помощью пакета **keyval**, который уже упоминался в главе 4 (Графика). В качестве параметров можно передавать выражения, если загружен пакет **calc**.

Список воспринимаемых пакетом опций очень обширен. В документации к пакету **geometry.pdf** подробное описание всех имеющихся опций занимает свыше пять страниц текста. Использование этого пакета для изменения размеров полей является предпочтительнее, чем изменяя размеров напрямую. Обилие принимаемых параметров позволят выразить любую идею по формату полосы набора в наиболее естественной форме, не ошибившись при расчётах. Подробное описание пакета к сожалению выходит за все разумные рамки на объём статьи. К счастью документация очень хорошая и чрезвычайно подробная.

**vmargin** Пакет **vmargin** управляет размерами макета через выставку полей. Перед выставкой полей следует задать формат листа:

---

```
\setpapersize [«ориентация»]{«формат листа»}
```

---

В качестве обязательного параметра команды можно указать один из следующих форматов: A0, A1, . . . , A9, B0, B1, . . . , B9, C0, . . . , C9, USletter, USlegal и USexecutive. Необязательный параметр может принимать значения **landscape** (альбомная ориентация) или **portrait** (портретная ориентация по умолчанию). Для не стандартных форматов листа можно задать размеры с помощью ключевого слова **custom** команды:

---

```
\setpapersize {custom}{«ширина»}{«высота»}
```

---

После того как удалось определиться с размером страницы можно задавать размеры для полосы набора с помощью одной из следующих команд:

---

```
%Полоса набора с колонтитулами
```

```
\setmargins{«ширина поля слева»}{«высота поля сверху»}%
           {«ширина текста»}{«высота текста»}%
           {«высота»}{«отступ»}%верхний колонтитул
           {«высота»}{«отступ»}%нижний колонтитул
```

```
%Полоса набора без колонтитулов
```

```
\setmarginohf{«ширина поля слева»}{«высота поля сверху»}%
              {«ширина текста»}{«высота текста»}%
```

```
%Установка размеров без изменения колонтитулов
```

```
\setmarg{«ширина поля слева»}{«высота поля сверху»}%
        {«ширина текста»}{«высота текста»}%
```

---

Кроме перечисленных в пакете **vmargin** определены их аналоги: `\setmarginrbr`, `\setmarginohfrb`, `\setmarginrb`. Отличие этих команд от вышеупомянутых заключается что в качестве параметров вместо ширины и высоты тела текста им передаются ширина правого поля и высота нижнего поля. Подробно пакет описан в документации **vmargin.pdf**.

### 6.3.2. Стили страницы

Полоса набора это не только размеры — это ещё и наполнение колонтитулов. Обычно наполнение колонтитулов определяется в классе документа. Лучше без необходимости ничего не менять.

Простейший способ изменить стиль страницы, это воспользоваться командой:

---

```
\pagestyle {«стиль страницы»}
```

---

Если стиль надо переопределить только для текущей страницы, то следует воспользоваться командой `\thispagestyle{стиль страницы}`.

Есть три стандартно определённых стиля:

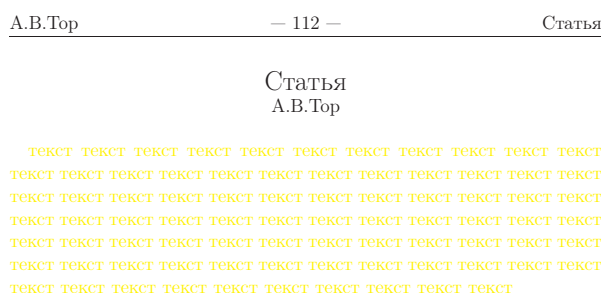
**empty** Страница выводится без каких-либо колонтитулов — только текст.

**plain** Выводится только номер страницы в нижнем колонтитуле.

**headings** В верхнем колонтитуле выводится номер страницы и информация, определяемая классом документа.

Если и это наполнение не устраивает, то можно определить свой собственный стиль. Пакет **fancyhdr** специализируется как раз на этом.

**fancyhdr** Чтобы воспользоваться возможностями пакета необходимо загрузить стилевой файл и с помощью `\pagestyle` выбрать стиль



Это просто демонстрация — схii — возможностей **fancyheadings**

Рис. 6.5. Заполняем колонтитулы с помощью **fancyhdr**.

Пакет **fancyhdr** позволяет управлять содержимым колонтитула. Возможности: верхний и нижний колонтитул разбивается на три независимых части, многострочные колонтитулы, колонтитулы, вылезаящие по ширине за `\textwidth`, декоративные линейки, разные колонтитулы для чётных и нечётных страниц, отдельные колонтитулы для специальных полос (начала глав, страницы, отведённые под плавающие объекты). Простейший пример для определения колонтитулов с помощью **fancyhdr**:

---

```

\usepackage{fancyhdr}
...
\begin{document}
\pagestyle{empty} %очищаем стиль страницы
\pagestyle{fancy} %включаем пользовательский стиль
\lhead{А.В.Топ} %верхний колонтитул слева
\chead{---~\arabic{page}~---} % там же по центру
\rhead{Статья} %верхний колонтитул справа
% аналогично для оформления нижнего колонтитула
\cfoot{---~\roman{page}~---}
\lfoot{\hspace{0.7cm}Это просто демонстрация}
\rfoot{возможностей \textbf{fancyheadings}}

```

---

Переменная `page` содержит в себе номер страницы. Подробности о том что можно сотворить с колонтитулами следует искать в документации к пакету `fancyhdr.pdf`.

## 6.4. Причёсываем текст

После того как границы определены посмотрим что можно сделать для красивого размещения текста на странице. Часть забот по следованию правилам вёрстки L<sup>A</sup>T<sub>E</sub>X берёт на себя. Вам, например, не требуется следить за единообразием оформления полос. Но есть дефекты которые компьютеру заметить не под силу, например, так называемые коридоры.

«Коридоры» графически разделяют текст абзаца или полосы на некоторое подобие неожиданных колонок, а они могут осложнять восприятие текста, из-за чего «коридоры» надо устранять как дефект набора.

Рис. 6.6. Дефект набора (коридор).

**Строка** Когда строка с точки зрения T<sub>E</sub>X становится слишком разряженной (Underfull) или слишком сжатой (Overfull), то в log-файле появляются предупреждения, начинающиеся со слов вида:

---

```

Overfull \hbox (26.1765pt too wide) in paragraph at lines 347--356
[[[[]\T2A/cmr/m/n/12 Список вос-при-ни-ма-е-мых па-ке-том оп-ций
очень об-ши-рен. В до-ку-мен-та-ции geometry.pdf

```

---

Если при выборе класса документа передать ему опцию `draft`:

---

```
\documentclass[ draft , a4paper , 12pt , oneside ]{ scrbook }
```

---

то такие проблемные места будут отмечаться прямо в тексте. Обычно, подобные

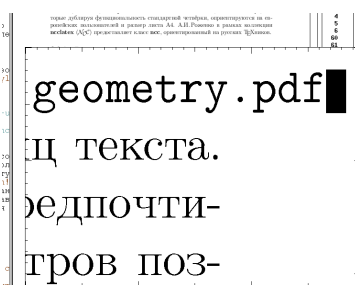


Рис. 6.7. Переполненная строка. Включена опция `draft`.

переполнения связаны с тем, что  $\text{\LaTeX}$  не знает как перенести какое-либо слово. В этом случае следует обучить его, что и где можно переносить, как это было показано в главе 2 (Базовые элементы). В крайнем случае можно насильно разорвать строку с помощью команды `\linebreak` или `\\`. В отличие от `\linebreak` команда `\\` не заставляет остаток строки выравниваться по правому полю.

Если можно редактировать текст, то для исправления дефектов набора лучше переделать предложение так, чтобы в новой инкарнации тест не создавал проблем для чтения.

**Горизонтальные пробелы** Расстояние между словами можно изменить с помощью горизонтальных промежутков. Горизонтальные промежутки создаются с помощью команды `\hspace`. В качестве параметра команде передаётся длина. Вариант команды `\hspace*` которые отличается от основной тем, что создание пробела не игнорируется даже тогда, когда пробел приходится на начало или конец строки.

Существует несколько определённых по умолчанию горизонтальных пробелов:

`\quad` — горизонтальный промежуток шириной `1em`. Так же есть `\qquad` — удвоенный `\quad`, и `\endspace` — половина от `\quad`.

`\hfill` — бесконечный горизонтальный промежуток. Два `\hfill` подряд в два раза больше, чем один. Так же есть более «маленькая» бесконечность — `\hfil`.

`\hrulefill` — то же, что и `\hfill`, но заполненный промежуток подчеркивается.

Аналогично есть команда заполняющая всё точками — `\dotfill`.

**Страница** Проблемы могут возникнуть и при формировании страниц. В крайнем случае можно воспользоваться командами принудительного завершения страницы `\pagebreak` или `\newpage`. Отличие первой команды от второй в том, что после формирования страницы полоса выравнивается по нижней кромке — это может привести к неоправданному растяжению страницы. Если проблему можно решить путём увеличения/уменьшения страницы на одну-две строки, то лучше воспользоваться следующими макросами:

---

```
\newcommand{\longpage}{\enlargethispage{\baselineskip}}
\newcommand{\shortpage}{\enlargethispage{-\baselineskip}}
```

---

Команда `\longpage` увеличивает тело текста текущей страницы на одну строку, а `\shortpage`, соответственно, уменьшает. Длина `\baselineskip` служит для определения *интерлиньяжа* — междустрочного пробела.

**Висячая строка** Одним из самых неприятных дефектов набора является «висячая строка». Висячая строка — это концевая строка абзаца, стоящая первой на странице, или начальная строка абзаца, стоящая на странице последней. Этих артефактов следует всячески избегать. Для подавления этого эффекта в заголовке документа следует определить две переменные:

---

```
%подавление висячих строк.
```

```
\clubpenalty=10000
```

```
\widowpenalty=10000
```

---

**Вертикальные просветы** По аналогии с командой `\hspace{длина}` вертикальные промежутки организуются с помощью команды `\vspace{длина}`. Модификация команды `\vspace*{длина}` создаёт вертикальный просвет, которые не игнорируется даже если просвет попадает на начало или конец страницы.

Вертикальные просветы так же имеют свои умолчания:

`\bigskip` — вертикальный промежуток равный примерно `\baselineskip`. Так же имеются `\medskip` — половина от `\bigskip`, и `\smallskip` — четверть от `\bigskip`.

`\vfill` — бесконечный вертикальный промежуток. Два `\vfill` подряд в два раза больше, чем один. Так же есть более «маленькая» бесконечность — `\vfil`.

**Печать через две строки** До сих пор временами встречаются требования вида: «Предоставить диплом, напечатанный через две строки» — пережиток эпохи печатных машинок. Для решение этой проблемы лучше всего воспользоваться пакетом **spacing**. В пакете определена команда `\doublespacing`, которая выполняет искомое действие. Так же в **spacing** определены макросы `\onehalfspacing` и `\singlespacing` — печать через полторы и одну строку, соответственно. Для вертикальной разрядки небольшого фрагмента текста лучше воспользоваться одноимёнными окружениями или окружением `spacing`:

```
\begin{spacing}{2.5}
  <<Этот текст, напечатан с
  интервалом в две с
  половиной строки>>.
\end{spacing}
```

«Этот текст, напечатан с интервалом в две с половиной строки».

В качестве основного параметра окружению `spacing` передаётся число строк через которое следует печатать текст.

## 6.5. Послесловие

В этом тексте присутствует далеко не вся информация необходимая при вёрстке текста. Несмотря на то, что  $\text{\LaTeX}$  позволяет верстать книги любителям без помощи профессионалов, но лучше при любой возможности спрашивать у этих профессионалов совета. Понимание *что, где и зачем* надо исправлять в случае  $\text{\LaTeX}$  чрезвычайно важно, потому что *как это следует сделать*, как правило, и так очевидно.