

L^AT_EX, GNU/Linux и русский стиль.

© Е.М. Балдин*



Эта статья была опубликована в июньском 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 ноября 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

Эмблемы T_EX и METAFont, созданные Дуайном Бибби, взяты со странички Д.Э. Кнута. Цветной пингвин взят из пакета ps2pdf от Ральфа Найпрашека (Rolf Niepraschk)

Оглавление

10. Всё о таблицах	1
10.1. Немного теории	1
10.2. tabbing	2
10.3. tabular и array	3
10.3.1. К вопросу о разделительных линиях	5
10.3.2. Клетки	5
10.3.3. Выравнивание чисел	7
10.3.4. Доступ к данным	7
10.3.5. Клоны tabular	7
10.4. Многополосные таблицы	8
10.5. Вывод	9
10.6. И это тоже таблицы?	10
10.7. В заключение о таблицах	10

Глава 10

Всё о таблицах

Классификация химических элементов, позволяющая выявить зависимость их различных свойств от атомной массы.

Таблица Менделеева — пожалуй, самая полезная из таблиц.

Таблица — это особая форма передачи содержания. Далеко не всякое содержание следует облекать в форму таблицы. Но для малых объёмов однородных значимых данных таблицы бывают лучшим способом отображения.

10.1. Немного теории

Содержимое таблицы организуется в колонки (графы) и горизонтальные строки таким образом, что каждый элемент является составной частью и строки, и колонки. Таблица состоит из следующих основных элементов: нумерационного и тематического заголовков (номер таблицы и её название), головки¹ (заголовочная часть таблицы), хвоста (вся остальная часть таблицы без головки), боковика (первая слева графа таблицы) и прографки (хвостовая часть таблицы без боковика).

Боковик	Прографка		Головка
ИЛИ	Истина	Ложь	
Истина	Истина	Истина	
Ложь	Истина	Ложь	Хвост

Таблица 10.1. Структура таблицы.

Заполняя таблицу текстовыми или цифровыми данными полезно следовать правилам:

¹В «издательском словаре-справочнике» А.Э. Мильчина утверждается, что попытка замены этого термина по той причине что он якобы является жаргонным не привилась.

- Выносить данные общие для каждого элемента графы в её заголовок, а общие для каждого элемента строки в заголовок боковика.
- По возможности употреблять числа не более чем из четырёх значащих цифр. Для этого более многозначные числа следует округлять. Общий множитель следует вынести в заголовок. То же самое нужно сделать и с единицами измерения.
- Всегда перед знаком, отделяющим целую часть числа от дробной, должна быть цифра. То есть правильно писать «0.1», но не «.1».
- Проставлять вместо отсутствующих данных многоточие «...», «Нет свед.». Если данных в принципе быть не может, то лучше отметить это с помощью тире «—».
- Не следует использовать знаки, означающее «то же, что и предыдущее значение». Лучше повторить число.

Честно говоря, описание таблиц с помощью ЛАТЭХ-разметки может показать не очень удобным. Действительно когда в перемешку идут данные и управляющие структуры результат может выглядеть не очень красиво. Спасает только то, что большие таблицы с однородной структурой можно создавать с помощью скрипта, а маленькие таблицы не являются особой проблемой. Как правило, гораздо больше времени чем оформление таблицы занимает поиск и верификация данных.

10.2. `tabbing`

Если точно известны ширина столбцов и таблица относительно простая, то можно воспользоваться окружением `tabbing`.

```
\begin{tabbing}
MMMM \= MMMM \=      \kill
\textbf{\ ИЛИ} \> Истина \> Ложь  \\
Истина \> Истина \> Истина \\
Ложь   \> Истина \> Ложь
\end{tabbing}
```

ИЛИ	Истина	Ложь
Истина	Истина	Истина
Ложь	Истина	Ложь

Первая строчка устанавливает положение табуляторов с помощью команды `\=`. Команда `\kill`, завершающая управляющую строку, даёт понять текстовому процессору, что её не надо печатать. Далее идёт обычный текст, где переход к следующей табуляции осуществляется с помощью команды `\>`, а перевод строки завершается комбинацией `\\`. Всё просто — точно так же таблицы печатали с помощью обычной печатной машинки, только табуляция задавалась «железным», а не программным способом.

К использованию окружения `tabbing` следует подходить с известной долей осторожности. Окружение формирует абзац, состоящий из отдельных строк, в котором

нет месту переносам и многострочным элементам. Зато это позволяет \LaTeX легко перенести часть таблицы на следующую страницу. Абсолютно всё приходится делать своими руками, но в этом есть своеобразная прелесть. Часто **tabbing** становится базой для определения других более специализированных окружений.

Некоторые стандартные команды \LaTeX внутри **tabbing** переопределены. Это в частности касается команд переноса ($\backslash-$) и акцентирования (\backslash' и $\backslash\grave{}$). Для сохранения/воспроизводства текущей табуляции следует воспользоваться командами $\backslashpushtabs/\backslashpoptabs$.

10.3. tabular и array

Самым популярным окружением для отображения таблиц в \LaTeX является **tabular**. Окружение **array** фактически полностью повторяет функционал **tabular**, но в отличие от последнего работает в математической моде — полезно для создания матриц, которые по своей сути тоже обычные таблицы.

```
\centering
\begin{tabular}[c]{c}{l|ll}
\textbf{\ ИЛИ} & Истина & Ложь \\ \hline
Истина & Истина & Истина \\
Ложь & Истина & Ложь
\end{tabular}
```

ИЛИ	Истина	Ложь
Истина	Истина	Истина
Ложь	Истина	Ложь

Данные делятся на ячейки с помощью символа «логическое И» $\&$. Переход на следующую строку контролируется стандартной командой переноса строки $\backslash\backslash$. В качестве необязательного параметра команды переноса строки можно указать дополнительный сдвиг по вертикали. Вертикальный размер каждой из строк автоматически выставляется в соответствии с высотой текста.

tabular создаёт единый объект — таблицу. Как и в случае картинок таблицу удобно заключать в плавающее окружение **table**:

```
\begin{table}[ht]
\centering%центрируем таблицу
\begin{tabular}[«позиционирование таблицы»]{«формат столбцов»}
«тело таблицы»
\end{tabular}
\caption{«подпись»}\label{tab:metka}
\end{table}
```

Это позволяет автоматически создать нумерационный заголовок таблицы и добавить тематический. Необязательный аргумент окружения **tabular** даёт возможность указать как позиционировать всю таблицу по вертикали по отношению к окружающему тексту:

t — выравнивание по верхней строке, то есть верхняя строка таблице будет расположена на одном уровне со строкой где эта таблица размещена,

- c** — выравнивание по центру,
- b** — выравнивание по нижней строке.

Ширина столбцов вычисляется автоматически по заданному формату, который задаётся через обязательный аргумент окружения. Каждому из столбцов должна соответствовать своя буква:

- l** (*left*) — выравнивание по левому краю,
- c** (*center*) — выравнивание по центру,
- r** (*right*) — выравнивание по правому краю,
- r**{«ширина»} — задание колонки определённой ширины. В случае жёстко заданной длины слишком длинный текст может разбиваться на несколько строк.

Если формат столбцов повторяется, то для сокращения записи можно воспользоваться следующей спецификацией:

`*{n}{«формат столбца или столбцов»}`

Где *n* — число повторений. Это своеобразный цикл.

```
\centering
\begin{tabular}{|*{3}{c|}}
\textbf{ИЛИ} & Истина & Ложь \\ \hline \hline
Истина & Истина & Истина \\
Ложь & Истина & Ложь
\end{tabular}
```

ИЛИ	Истина	Ложь
Истина	Истина	Истина
Ложь	Истина	Ложь

Разделительные линии между столбцами задаются с помощью вертикальной черты `|`. Две вертикальные линии `||` формируют двойной разграничитель. Горизонтальные линии создаются с помощью команды `\hline`. По аналогии с двойной вертикальной чертой две команды формируют двойную горизонтальную линию. Инструкция `@{}` позволяет вставить между столбцами любой символ указанный в качестве обязательного аргумента. При этом подавляются околостолбцовые промежутки, добавляемые в по умолчанию автоматически. Это можно быть полезно в случае если один столбец представляет из себя какую-то измеренную величину, а второй её ошибку — в этом случае вместо разделительной черты между ними лучше вставить знак \pm .

```
\centering
\begin{tabular}{c|p{2cm}@{${\pm}$}r|}
\textbf{ИЛИ} & Истина & Ложь \\ \cline{2-3}
Истина & Истина & Истина \\ \cline{1-1}\cline{3-3}
Ложь & Истина & Ложь \\ \cline{2-2}
\end{tabular}
```

ИЛИ	Истина	\pm	Ложь
Истина	Истина	\pm	Истина
Ложь	Истина	\pm	Ложь

Для того чтобы отчеркнуть только часть столбцов можно воспользоваться командой `\cline{диапазон столбцов}`.

Окружение `array` в дополнение к стандартным типам столбцов, используемых в `tabular`, добавляет два новых:

`m{«ширина»}` — то же, что и `r{«ширина»}`, но добавляется вертикальное выравнивание содержимого по центру клетки ,

`b{«ширина»}` — то же, что и `m{«ширина»}`, но вертикальное выравнивание содержимого идёт по нижней базовой линии последней строки.

Более подробную информацию об использовании окружения `array` следует искать в файле документации `array.pdf` из пакета `tools`.

Подробнее о том как *должна* выглядеть таблица и как это достигается можно узнать, например, в статье «Publication quality tables in L^AT_EX» (`booktabs.pdf`), написанной Симоном Фиром (Simon Fear) для пакета `booktabs`. Этот пакет для тех, кто любит везде наводить лоск.

10.3.1. К вопросу о разделительных линиях

По характеру оформления линейками таблицы бывают закрытые (глухие), полузакрытые, открытые. Два настоятельных совета:

- Никогда не следует использовать вертикальные линии в таблице.
- Двойные линии в оформлении таблицы также лишние.

Если данные в таблице настолько разные, что хочется их поделить вертикальной линией, то лучше сделать две таблицы. Но опять же никто не может запретить использовать разделительные линии.

Стиль `hline` из коллекции `tools` определяет команду `\hline`, которая позволяет управлять созданием двойной рамки, не создавая ненужных пересечений. Подробности в документации `hline.pdf`.

Пакет `arydshln` необходимо в случае использования пунктирных разделительных линий. Документация к этому пакету `arydshln-man.pdf` более чем исчерпывающая.

10.3.2. Клетки

Для объединения рядом расположенных по горизонтали клеток можно воспользоваться командой `\multicolumn{n}{формат колонки}{текст}`:

```
\centering
\begin{tabular}{|*{3}{c|}}
\textbf{ИЛИ} & Истина & Ложь \\ \hline
Истина & \multicolumn{2}{c}{Истина} \\
Ложь & Истина & Ложь
\end{tabular}
```

ИЛИ	Истина	Ложь
Истина	Истина	
Ложь	Истина	Ложь

Первый обязательный параметр соответствует числу объединённых колонок, второй — формату получившейся объединённой колонки, третий — текст.

Для объединения клеток по вертикали можно воспользоваться пакетом **multirow** в котором определяется набор одноимённых команд:

```
\centering
\begin{tabular}{|*{3}{p{1.2cm}}|}
\textbf{ИЛИ} & Истина & Ложь \\ \hline
Истина & \multirow{2}{Истина} & Истина \\
Ложь & & Ложь
\end{tabular}
```

ИЛИ	Истина	Ложь
Истина	Истина	Истина
Ложь	Ложь	Ложь

Подробнее об использовании команд пакета написано в README к нему. В пакете определены две команды с двумя и тремя обязательными аргументами:

```
\multirow {«число строк»}{«ширина»}{«текст»}
\multirow {«число строк»}*{«текст»}
```

В качестве первого аргумента передаётся число строк, которые займёт объединённая клетка, далее можно выбрать либо автоматическое вычисление ширины, или указать её самостоятельно. Число строк может быть отрицательным. В этом случае объединяются ячейки сверху от команды. Подобное может потребоваться, чтобы согласовать свою работу с пакетом **colortbl**:

```
\centering
\begin{tabular}{c>{\columncolor{yellow}}cc}
\backslashbox{два}{один} & Истина & Ложь \\
Истина & & Истина \\
Ложь & \multirow{-2}{Истина} & Ложь
\end{tabular}
```

	один	Истина	Ложь
два	Истина	Истина	Истина
	Ложь	Ложь	Ложь

Пакет **colortbl** предназначен для раскрашивания таблицы в разные цвета. В документации **colortbl.pdf** подробно излагаются принципы работы пакета.

Команда `\backslashbox{текст}{текст}`, делящая клетку на две части наклонной чертой, определена в пакете **slashbox**. Иногда так оформляют клетку на перекрестии боковика и головки. С пакетом идёт пример **slashbox.tex**.

Автор кириллических шрифтов **lh** Ольга Лапко для более изошрённой работы с таблицами предлагает пакет **makecell**. В это пакете определена команда `\makecell`, которая «создаёт окружение одноклоночной таблицы с предопределёнными общими параметрами выключки, интерлиньяжа и вертикальных отбивок вокруг. Её удобно использовать для многострочных ячеек. Дополнительный аргумент команды позволяет изменить выключку таблицы». В пакете документация на английском языке (**makecell.pdf**) дублируется русской документацией (**makecell-rus.tex**).

10.3.3. Выравнивание чисел

Таблица часто строится вокруг чисел. Поэтому не удивительно, что внимание к выравниванию чисел для целей упрощения восприятия данных должно быть повышенным.

Пакет **dcolumn** из коллекции **tools** добавляет ещё одну спецификацию к формату столбцов таблицы:

```
\centering
\begin{tabular}{|c|l|D{.}{,}{5}|}
0.3141 & 0.3141 & 0.3141 \\
3.141 & 3.141 & 3.141 \\
31.415 & 31.415 & 31.41 \\
\end{tabular}
```

0.3141	0.3141	0,3141
3.141	3.141	3,141
31.415	31.415	31,41

Новая спецификация имеет формат: $D\{\text{delim}\}\{\text{output}\}\{\text{nfrac}\}$, где *delim* — символ или набор символов по которому происходит выравнивание (обычно это точка или запятая), *output* — символ который замещает *delim* при компиляции (например, бывает нужно заменить точку на запятую), *nfrac* — максимальное число позиций в дробной части числа (при отрицательном значений число позиций не фиксируется). Подробности в документации **dcolumn.pdf**.

Пакет **rccol** обладает схожей функциональностью, что и **dcolumn**, но дополнительно позволяет округлять значения. К сожалению мне не удалось заставить его нормально работать если в качестве разделителя используется точка. **rccol** не смотря на информацию в документации **rccol.pdf** признаёт только запятую. Возможно, это недоразумение.

Пакет **warpcol** предоставляет общую процедуру формирования формата числовых колонок. В документации к пакету **warpcol.pdf** представлены примеры как добиться различных эффектов при выравнивании.

10.3.4. Доступ к данным

CSV (от англ. Comma Separated Values значения, разделённые запятыми) — это текстовый формат, предназначенный для представления табличных данных. Для доступа к этим данным напрямую можно воспользоваться пакетом **csvtools**. Документация к пакету **csvtools.pdf** достаточно подробна, но лучше к данным доступаться с помощью скрипта результатом действия которого является сам *tex*-файл.

10.3.5. Клоны tabular

tabularx из коллекции **tools** — расширение **tabular** на предмет автоматического вычисления ширины колонок, имеющих форматный определитель *X*.

```

\centering
\begin{tabularx}{\textwidth}{|D{.}{,}{4}|X|c|}
0.3141 & очень длинная строка & 0.3141 \\
3.141\footnote{В окружение tabularx можно
добавлять подстрочное примечание}
& 3.141 & 3.141
\end{tabularx}

```

0,3141	очень длинная строка	0.3141
3,141 ^a	3.141	3.141

^aВ окружение tabularx можно добавлять подстрочное примечание

В качестве первого аргумента окружения **tabularx** передаётся ширина таблицы. При компиляции X преобразуется в $r\{\text{ширина}\}$, где вместо ширины выставляется автоматическим образом вычисленная величина так чтобы в целом ширина таблицы оставалась неизменной. Если в таблице присутствует более одного столбца, имеющих формат X , то для формирования таблицы может потребоваться несколько проходов. Все подробности, как обычно, можно узнать в документации к пакету (`tabularx.pdf`).

Пакет **tabulary** так же является модификацией стиля **tabular**, точнее **array**. Пакет был создан для автоматического расчёта ширины колонок таблицы с целью минимизации высоты таблицы. Подробности в документации к пакету `tabulary.pdf`. Пользоваться одноимённым окружением следует с некоторой долей осторожности.

ctable альтернативный взгляд на оформление таблицы. Вместо окружения используется команда `\ctable`, которая объединяет в себе функциональность **tabular** и **table**. В дополнение этим особенностям в `\ctable` можно вставлять команды для создания подстрочных примечаний для таблицы. Примеры и документацию к пакету можно найти в файле `ctable.pdf`.

10.4. Многополосные таблицы

Окружение **tabular** и его производные всем хороши за исключением того, что они не могут занимать больше одной страницы. Для вёрстки на несколько страниц обычно используют одно из двух окружений: **supertabular** и **longtable** определённых в одноимённых стилях. Оба эти окружения обладают схожей функциональностью, но разными способами оформления. Оба предоставляют возможность создания стандартной шапки и стандартного окончания появляющихся в начале, на каждой новой странице и в конце таблицы. Подробности можно отыскать в документации к пакетам: `supertabular.pdf` для **supertabular** и `longtable.pdf` для **longtable** из коллекции **tools**.

longtable в отличие от **supertabular** гарантирует неизменность ширины столбцов на протяжении всей таблицы, что достигается за несколько проходов \LaTeX . Неизменность ширины столбцов в зависимости от ситуации может рассматриваться как преимущество, так и как недостаток.

В пакете **supertabular** кроме окружения **supertabular** определено окружение **mpsupertabular**, которое каждый отдельный кусок таблицы на своей странице

заключает в окружение **minipage**, что позволяет использовать подстрочные примечания прямо в таблице.

Более молодым и, возможно, более правильным является пакет **xTAB**. По сути дела это расширение **supertabular** с некоторыми улучшениями и исправлениями. Подробная документация доступна в файле `xTAB.pdf`

10.5. Вывод

Вывод — таблица без линеек или организованный в колонки и строки материал, который не разделён линейками. Оглавление `\tableofcontents`, списки иллюстраций `\listoffigures` и таблиц `\listoftables` тоже являются таблицами.

```
\centering
\begin{tabular}{p{0.7\textwidth}@{r}}
Глава 1\dotfill & 2 \\
Глава 2\dotfill & 10 \\
Глава 3\dotfill & 124
\end{tabular}
```

Глава 1	2
Глава 2	10
Глава 3	124

Команда `\dotfill` формирует *отточия* (заполнение пространства точками).

Окружение **listliketab** из одноимённого пакета специализируется на создании таблиц, оформленных как перечисление. Это окружение будет очень кстати при оформлении вопросников:

```
\storestyleof{enumerate}
\begin{listliketab}
\newcounter{tabenum}\setcounter{tabenum}{0}
\newcommand{\nextnum}{\addtocounter{tabenum}{1}\thetabenum.}
\begin{tabular}{L>\bf}l@{~или~}>\bf}l@{~или~}>\bf}l}
\nextnum & Красный & & зелёный & & голубой \\
\nextnum & Низкий & & средний & & высокий \\
\nextnum & Радостный & & грустный & & удивлённый \\
\end{tabular}
\end{listliketab}
```

1. **Красный** или **зелёный** или **голубой**
2. **Низкий** или **средний** или **высокий**
3. **Радостный** или **грустный** или **удивлённый**

Документация с примерами находится в файле `listliketab.pdf`

