

Пакет `paralist`*

Расширение для окружений перечней

Bernd Schandl
schandl@gmx.net

Printed 1 октября 2008 г.

Аннотация

Данный пакет предлагает новые окружения перечней. Маркированные и нумерованные перечни могут быть напечатаны внутри абзаца, как несколько абзацев и в сжатом макете. Большинство окружений имеют опции форматирования меток. Дополнительно окружения ЛАТЭХ'a `itemize` и `enumerate` могут быть модифицированы, при использовании опции задания меток.

Техническое редактирование¹

Задание нумерованных перечней по умолчанию (<code>\setdefaultenum</code>) 5	Компактные перечни маркированные (<code>compactitem</code>) 7
Задание маркированных перечней по умолчанию (<code>\setdefaultitem</code>) 5	нумерованные (<code>compactenum</code>) 7 перечни-описания (<code>compactdesc</code>) 7
Задание полей для меток по умолчанию (<code>\setdefaultleftmargin</code>) 5	

Содержание

1	Введение	2
2	Опции пакета	2

*Пакет версии v2.3b of 2002/03/18.

¹Добавлено при переводе.

3	Оформление меток	3
4	Значения по умолчанию для меток и полей	5
5	Новые окружения перечней	6
5.1	Нумерованные окружения перечней	6
5.2	Маркированные окружения перечней	7
5.3	Окружения перечней-описаний	7
5.4	Перечни с пустыми метками	8
6	Вложения перечней	8
7	Настройки	9
8	Недочёты и пожелания	9
9	Acknowledgments	10

1 Введение

В рассылку `comp.text.tex` в мае 1998, кто-то прислал письмо с вопросом об окружении перечня которое (a) может использоваться внутри абзаца, (b) следить за нумерацией и (c) иметь метки, на которые можно сослаться. В другом письме указали на пакет `theapa` как возможное решение. Сейчас, глядя на это окружение и найдя эти старые письма, я посмотрел на пакет `theapa` и решил кое-что позаимствовать и немного переписать свой пакет.

Помимо этого были добавлены компактные версии `enumerate`, `itemize` и `description` и заданы опции для большинства окружений для возможности специального форматирования меток.

2 Опции пакета

Если определены опции, то подключаются соответствующие части пакета.

`newitem/olditem` При `newitem` (заданной по умолчанию), окружения ЛАТЭХ'a `itemize` будут переопределены с тем, чтобы задавать опции для формата метки. См. разд. 3. При задании `olditem` `itemize` остаётся как есть.

newenum/oldenum При `newenum` (заданной по умолчанию), окружения \LaTeX 'а `environment enumerate` будут переопределены с тем, чтобы задавать опции для формата метки. См. разд. 3. При задании `oldenum enumerate` остаётся как есть.

alwaysadjust Ширина меток перечней `compactenum`, `enumerate`, `compactitem` и `itemize` всегда пересчитывается на размер метки. Для меток по умолчанию, это означает, что ширина метки обычно уменьшается.

neveradjust Ширина меток не пересчитывается, в том числе и для меток, заданных вручную. Эта опция игнорируется если задана `alwaysadjust`.

neverdecrease Если ширина меток пересчитывается, данная опция отменяет сужение ширины. Пример, когда это может понадобиться. Если не задано `...adjust`, то втяжка `\item` в `\begin{enumerate}` и `\begin{enumerate}[1.]` будет разной, даже если у них одинаковые метки.

defblank Два окружения `inparablank` и `asparablank` будут определены. См. разд. 5.4.

pointlessenum Метки в нумерационных перечнях будут помечаться и создавать ссылки вида «1», «1.1», «1.1.1» и «1.1.1.1». См. также разд. 3.

pointedenum Метки в нумерационных перечнях будут помечаться ссылки вида «1.», «1.1.», «1.1.1.» and «1.1.1.1.» и создавать ссылки без последней точки. См. также разд. 3. Эта опция игнорируется, если задана `pointlessenum`.

flushright Метки в четырёх перечнях, указанных выше, выравниваются по правому краю. Так как это задано в \LaTeX 'е по умолчанию, это также задано по умолчанию и в этом пакете.

flushleft Метки в четырёх перечнях, указанных выше выключаются влево.

cfg Загружается, если существует, файл `paralist.cfg` (по умолчанию)

nocfg Конфигурационный файл не загружается.

Опция `increaseonly` убрана; используйте `neverdecrease instead`.

3 Оформление меток

Все маркированные и нумерованные окружения имеют опции для задания формата метки. Следующие примеры будут работать если `paralist` загружен *без* опций `olditem` и `oldenum`.

В стандартных классах \LaTeX 'а, окружение `itemize` использует сле-

дующие знаки для своих четырёх уровней: • – * ·. Если вы хотите их изменить только в отдельном окружении, задаёте

```
\begin{itemize}[$\star$]
```

и в перечне будет использоваться знак *.

Опции нумерованного перечня несколько более сложные, но кто работал с пакетом `enumerate` (автор David Carlisle) может пропустить этот раздел, поскольку используется (почти) такой же механизм.

Если вхождения (токены) типа `A`, `a`, `I`, `i`, `1` заданы в опции, то используются соответствующие стили счётчиков `\Alph`, `\alph`, `\Roman`, `\roman` и `\arabic`. Эти буквы/цифры могут быть окружены текстом или кодом Т_ЕX'a. Но если токены `A` `a` `I` `i` `1` не должны выполнять роль счётчиков, они должны быть взяты в группу `{ }`. Несколько примеров.

```
\begin{enumerate}[(i)]
```

создаст метки (i), (ii), (iii) ...

```
\begin{enumerate}[{example} a)]
```

создаст a), example b), example c) ...

```
\begin{enumerate}[{A}-1]
```

создаст A-1, A-2, A-3 ...

```
\begin{enumerate}[\bfseries {Item} I]
```

создаст **Item I**, **Item II**, **Item III** ...

Обратите внимание, что в последнем примере `[\textbf{Item} I]` не станет работать, так как токен обозначения нумерации, `I`, попал в группу.

Команда `\ref` создаёт только счётчик без окружающего текста, поэтому из примеров выше вы получите `i`, `a`, `1` и `I` для перечней верхнего уровня.

`\pointedenum` Есть также две опции пакета и две макрокоманды, создающие метки и ссылки на них. Опция `pointedenum` и команда `\pointedenum` создают метку «1.», «1.1.», «1.1.1.» и «1.1.1.1.» ссылки без последней точки. Опция `pointlessenum` и команда `\pointlessenum` вообще не задаёт последней точки.

В то время как опции имеют глобальное значение, команды могут использоваться для локальных изменений перечня, например

```
\newenvironment{myenum}%  
  {\pointedenum\begin{enumerate}}%  
  {\end{enumerate}}
```

Обратите внимание, что `\begin{enumerate}` задаётся *после* `\pointedenum`, иначе будет невозможно задать опцию.

`\paradescriptionlabel` В классах документа, формат метки `description` определяется как `\newcommand*\descriptionlabel[1]{%`

```
\hspace\labelsep\normalfont\bfseries #1}.
```

Это также используется и для `compactdesc`. Для окружений `inparadesc` и `asparadesc`, создана отдельная команда `\paradescriptionlabel` которая определена (фактически) так:

```
\newcommand*\paradescriptionlabel[1]{%  
  \normalfont\bfseries #1}.
```

4 Значения по умолчанию для меток и полей

Если вы хотите переопределить в документе перечни, заданные L^AT_EX'ом по умолчанию, будет несколько странным задавать их каждый раз в опции. Поэтому предлагаются три команды переопределения меток и полей перечней.

Обратите внимание, что макрокоманды, определяющие метки *не* адаптируют поля, поскольку это может вызвать побочные эффекты. Если вы захотите сделать такие изменения, используйте специальную команду `\setdefaultleftmargin`.

Если один из аргументов команды пустой, метка или поле этого уровня остаются без изменений.

`\setdefaulttitem` Метки по умолчанию для маркированных перечней задаются командой `\setdefaulttitem` которая использует четыре аргумента. Чтобы получить метки L^AT_EX'a, заданные по умолчанию, запишите

```
\setdefaulttitem{\textbullet}%  
  {\normalfont\bfseries \textendash}%  
  {\textasteriskcentered}{\textperiodcentered}
```

(это несколько глупо, поскольку ничего не нужно делать, чтобы использовать метки, заданные по умолчанию). Если вам нужен треугольник (\triangleright) вместо тире во втором уровне, задайте

```
\setdefaulttitem{}{\$ \triangleright \$}{-}{-}.
```

`\setdefaultenum` Метки нумерованных перечней задаются командой `\setdefaultenum`, используя механизм, описанный в разд. 3. Метки L^AT_EX'a по умолчанию могут быть определены как

```
\setdefaultenum{1.}{(a)}{i.}{A.}.
```

Если вам нужны прописные римские цифры для третьего уровня, задайте

```
\setdefaultenum{}{}{I.}{}.
```

`\setdefaultleftmargin` Для изменения полей для перечней, используется `\setdefaultleftmargin`.

Длина `\leftmargin n` определяет втяжку уровня n которая учитывает втяжку верхнего перечня $n - 1$ или текста (если $n = 1$)[???]. Окру-

жения, использующие `\leftmargin n` — это (как минимум) `enumerate`, `compactenum`, `itemize` и `compactitem` (может и больше, всех не упоминай). Установки ЛАТЭХ'a следующие

```
\setdefaultleftmargin{2.5em}{2.2em}{1.87em}{1.7em}{1em}{1em}.
```

В режиме `twocolumn` ЛАТЭХ использует меньшие поля для первого, пятого и шестого уровней, что может определено как

```
\setdefaultleftmargin{2em}{}{}{.5em}{.5em}.
```

Команды `\defaultitem`, `\defaultenum` и `\defaultleftmargin` не используются. Они остались для обратной совместимости с версиями < 2.1.

Если какие-то изменения переходят из документа в документ, при использовании `paralist`, поместите их в файл `paralist.cfg` который будет прочитан в конце загрузки пакета (только если вы не задали опцию `nosfg`).

5 Новые окружения перечней

5.1 Нумерованные окружения перечней

`asparaenum` Окружение `asparaenum` аналогично нумерованному перечню, но метки оформлены как отдельные абзацы.

Как пример, используем `asparaenum` внутри этого абзаца.

1. Каждый пункт `\item` в основном задаётся как отдельный абзац. Вторая строка *не* имеет втяжки (это особенность, а не ошибка).
2. Следующий пункт `\item` выглядит как этот и имеет метку.

Пример содержит следующий код:

```
\begin{asparaenum}
  \item Каждый ...
  \item Следующий ... \label{p11}
\end{asparaenum}
```

Задав `\ref{p11}` мы получим [2](#).

`inparaenum` Окружение `inparaenum` создаёт перечень внутри абзаца, аналогично тому, что был задан во введении.

Пример во введении был задан следующими командами:

```
... об окружении перечня которое
\begin{inparaenum}[\quad(a)]
  \item может использоваться внутри абзаца,
  \item следить за нумерацией и
  \item иметь метки, на которые можно сослаться. \label{p12}
```

```
\end{inparaenum}
В~другом письме указали ...
```

Задав `\ref{p12}` получим `c`.

`compactenum` Окружение `compactenum` является компактной версией стандартного окружения `enumerate`. Все вертикальные отбивки обнулены (но их можно отредактировать, см. разд. 7).

5.2 Маркированные окружения перечней

`asparaitem` Окружение `asparaitem` аналогично `asparaenum`. Оно использует символы вместо нумерации. Окружение может иметь опцию где задаётся символ. Для примера см. разд. 6.

`inparaitem` Аналогично `inparaenum` было добавлено окружение `inparaitem`, в котором также можно использовать опции. Я, в действительности, не знаю кому это может понадобиться и добавил его только для симметрии.

`compactitem` Окружение `compactitem` опять же компактная версия стандартного `standard itemize` с обнулёнными отбивками. Итак создадим окружение

- вы можете сэкономить место
- определить символ.
- Позвольте задать более длинный пункт, чтобы вы увидели, что здесь другая втяжка, чем у окружения `asparaitem`.

А код примера выше такой

```
\begin{compactitem}[$\circ$]
\item вы можете сэкономить место
\item определить символ.
\item Позвольте задать ...
\end{compactitem}
```

5.3 Окружения перечней-описаний

`asparadesc` Окружение `asparadesc` опять похоже на `asparaenum`. Использует опцию команды `\item` как «вступление» в абзаце.

`inparaitem` Аналогично `inparaenum`, я добавил `inparadesc`. Возможно никому оно не нужно и опять добавлено для симметрии.

`compactdesc` Окружение `compactdesc` скопировано из стандартных классов ЛАТ_EX'a с уничтоженными отбивками. Вопрос между делом, знает ли кто-нибудь, почему `description` определена в классе документа, а не в файле ядра `ltlists.dtx`?

5.4 Перечни с пустыми метками

Иногда требуется создать перечень с метками, как будто перечня нет. Кажется, так легче поместить структурированные данные в `LyX`. Поскольку это нужно не всем эти (странные) окружения определяются, если задана опция `defblank`. Следующие два окружения не имеют опций, поскольку нет меток, которые нужно оформлять.

`asparablanк` Каждая метка оформлена в виде обычного абзаца. Если вы хотите использовать опцию в команде `\item`, вам нужно добавить пробел в конце метки, поскольку `\labelsep` обнулён. Используйте что-то типа `\item[\textbullet\hspace{.5em}]`

`inparablanк` Метки задаются как регулярный текст. «Проблема пробела» упомянутая в предыдущем абзаце решается автоматически. Если бы я вам не сказал, то вы бы и не знали, что этот абзац был набран так:

```
... задаются
\begin{inparablanк}
  \item как ...
  \item «Проблема» ...
  \item решается ...
\end{inparablanк}
Если бы ...
```

6 Вложения перечней

Все окружения могут быть вложены одно в другое, как и стандартные окружения перечней хотя результаты не всегда будут те, что ожидаются. Например, неплохая идея вызвать другой перечень внутри окружения `inpara...`, неужели это никому не понадобится? Максимальный уровень вложения равен шести (и четырём для одного вида), как и для окружения `LaTeX`'а.

Этот абзац

★ пример использования `asparaitem` вместе с опцией,

★ и демонстрация что (i) можно использовать `inparaenum` внутри `asparaitem` и (ii) делать ссылки на пункты.

Ссылка была сделана в пункте (ii). Код последнего примера

```
\begin{asparaitem}[$\star$]
  \item пример ...
  \item и~демонстрации
    \begin{inparaenum}[(i)]
      \item можно использовать ...
```

```

\item делать ссылки ... \label{p13}
\end{inparaenum}
\end{asparaitem}
Ссылка была сделана в~пункте (\ref{p13}).

```

7 Настройки

Настройки, я уже слышал что-то вроде «Ваши компактные перечни хорошая идея, но мне хотелось бы сделать их менее компактными.» Пожалуйста, решение проблемы. Следующие отбивки могут быть изменены с использованием `\setlength` и создать отбивки в `compact...` 'ных окружениях. Имена отбивок похожи на имена L^AT_EX'a, поэтому я просто копирую объяснение из `ltlists.dtx`.

`\pltopsep`: Отбивка между первым пунктом и предыдущим абзацем.

`\plpartopsep`: Дополнительная отбивка к `\topsep` когда окружение начинается новый абзац (в так называемой вертикальной моде).

`\plitemsep`: Отбивка между пунктами.

`\plparsep`: Отбивка между абзацами внутри пункта — параметр `\parskip` для данного окружения.

Фактически, две отбивки `...topsep` добавляются до *и после* перечня.

Значение по умолчанию для всех них равно 0 pt. Возможно неплохая идея определить их через параметры шрифта, т. е. используя единицы `ex` или `em`.

8 Недочёты и пожелания

Недочётов нет ... о которых я знаю.

Итак, фактически один пункт, если вы используете пакет `babel` с одной из опций `acadian`, `canadien`, `francais`, `frenchb` или `french` (что фактически одно и то же). То, поскольку они переопределяют окружение `itemize` в начале документа, определение `itemize` из пакета `paralist` теряется. Есть три возможных решения:

1. Использовать окружение `itemize` без опции. :(
2. Использовать `\FrenchItemizeSpacingfalse` после загрузки `babel`, что отменит переопределение `itemize` пакетом `babel`. :/
3. Создайте вариант скомбинированного кода из пакетов `babel` и `paralist` и пришлите его мне. :)

Не стесняйтесь дать мне знать о проблемах, предложениях и пожеланиях по совершенствованию пакета и документации. Вознагражде-

ния также приветствуются ;-). Последнюю версию пакета всегда можно найти на CTAN'е или на <http://schandl.gmxhome.de/paralist/>.

9 Acknowledgments

I want to thank all the users who helped me with their comments finding bugs and extending the package. A big «Thank you» goes to David Carlisle, because there wouldn't be any optional arguments for the enumerated environments without the code from his `enumerate` package. Some pieces of code of the `inpara...` environments are inspired by Mogens Lemvig Hansen's `shortlst` package.