

Инструкции по подготовке сборника трудов конференции в стиле mmro.sty

Воронцов К. В.

voron@ccas.ru

Москва, Вычислительный центр РАН

Этот документ является руководством по стилевому файлу `mmro.sty` и предназначен для редактора сборника трудов конференции.

Команды для объединения статей в сборник

Каждая статья содержится в отдельном L^AT_EX-файле, который может быть скомпилирован как отдельно, так и в составе сборника. При-сылаемые авторами статьи не нуждаются ни в какой специальной правке, чтобы их можно было вставить в сборник. Единственное требование к статьям — они должны подключать стиль `mmro.sty` и корректно компилироваться по отдельности.

► `\begin{papers} ... \end{papers}`

Главный L^AT_EX-файл, который компилируется в сборник, должен содержать окружение `papers`, внутри которого каждая статья подключается командой `paper`. Пример:

```
\begin{papers}
  \paper+--+{ivanov}[scorr,voron]
  \paper+++{petrov}
  \paper++!{sidorov}[andrew]
\end{papers}
```

Вне окружения `papers` команда `paper` не работает.

Стиль `mmro.sty` предполагает, что все статьи находятся в одной директории. Поэтому имя каждого файла должно быть уникально. Удобнее всего именовать файлы по фамилии первого автора статьи. Вставляемые графические файлы также должны иметь уникальные имена. Проблема возникнет, когда два автора назовут свои графические файлы одинаково, скажем, `Fig1.eps`. Конфликт устраняется путём переименования графического файла (разумеется, и в тексте статьи тоже).

► `\paper $\alpha\beta\gamma$ {имя файла} [корректоры, рецензенты]`

Включить в сборник статью с указанным *именем файла*. Имя файла не должно содержать русских букв. Расширение `.tex` указывать не нужно.

Символ α может принимать два значения: + если статья уже прошла корректуру, - если ещё не прошла.

Символ β может принимать два значения: + если статья уже прошла рецензирование, - если ещё не прошла.

Символ γ может принимать четыре значения: + если статья принята в сборник, ? если статья принята в сборник, но желательна её авторская доработка, ! если статья не может быть принята без доработки, - если статья не принята.

В необязательном аргументе указывается список назначенных для данной статьи *корректоров и рецензентов*, через запятую. Имена (псевдонимы) корректоров и рецензентов не должны содержать русских букв.

Новые поступающие статьи рекомендуется включать в сборник командой `\paper--{имя файла}`. Место вставки выбирается редактором вручную, например, в каждом разделе сборника в алфавитном порядке по фамилии первого автора.

► `\chapter{код раздела}{расшифровка}{название раздела}`

Начало нового раздела (главы) в сборнике. Раздел всегда начинается с новой страницы, причём обязательно с нечётной, и обязательно без колонтитулов. *Название раздела* печатается крупным шрифтом на новой странице, под ним мелким шрифтом печатается *код раздела* и его *расшифровка*. Код раздела — это аббревиатура, которая будет выводиться в колонтитулах, пока идёт данный раздел.

► `\chaptercode{код раздела}`

Команда задаёт код раздела, который будет выводиться в колонтитулах на всех последующих страницах. Команда с пустым аргументом `\chaptercode{}` подавляет вывод кода раздела в колонтитулах.

► `\FILTER $\alpha\beta\gamma$ [корректор или рецензент]`

Эта команда задаёт условие отбора всех последующих статей в сборник. Например, можно собрать сборник из всех статей, которые уже прошли корректуру и рецензирование, или из всех не принятых статей, или из всех статей, назначенных данному рецензенту. Эти возможности полезны на промежуточных стадиях подготовки сборника.

Набор символов $\alpha\beta\gamma$ имеет тот же смысл, что и в команде `\paper`. При включении статей в сборник символ $\gamma = ?$ приравнивается к +, символ $\gamma = !$ приравнивается к -.

В команде `\FILTER` любой из трёх символов $\alpha\beta\gamma$ может быть задан как *. Это означает, что статьи включаются в сборник независимо от значения данного символа.

Необязательный аргумент позволяет задать одного *корректора или рецензента*. Здесь, в отличие от команды `\paper`, список корректоров

и рецензентов недопустим. Если необязательный аргумент не задан, будут выводиться все статьи. Если задан пустой необязательный аргумент [], будут выводиться все статьи, для которых список корректоров и рецензентов не задан или пуст.

По умолчанию в черновом режиме `draft` установлен `\FILTER***` (включать все статьи), а в обычном режиме `\FILTER**+` (включать только принятые к публикации).

► `\FINALSTAT`

Команда выдаёт итоговую статистику статей. Её имеет смысл вставлять один раз после `\end{papers}` на промежуточных стадиях подготовки сборника. Выдаётся следующая информация:

```
Всего подано статей: ...
Статей, прошедших корректуру: ...
Статей, прошедших рецензирование: ...
Всего статей с \REVIEWERNOTE: ...
(всего сделано замечаний: ...)
Всего статей с \FINALREVIEWERNOTE: ...
Статей, отмеченных как принятых в печать: ...
Статей, которые желательно доработать: ...
Статей, которые не могут быть опубликованы без доработки: ...
Статей, которые уже точно отброшены: ...
Статей в этой версии сборника: ...
Всего иллюстраций в этой версии сборника: ...
```

Команда `\FINALSTAT` работает только в черновом режиме `draft`.

Опция чернового режима `draft`

При включении данной опции `\usepackage[draft]{mmro}` в текст сборника выводится следующая дополнительная информация:

- выводятся все статьи, включая непринятые, если командой `\FILTER` не задано иное условие отбора;
- в статьях выводятся замечания рецензентов;
- в содержании для каждой статьи выводится: все авторы и название статьи, имя файла, список корректоров и рецензентов, текущее состояние статьи (расшифровка символов $\alpha\beta\gamma$), количество замечаний рецензентов, имена всех графических файлов, включённых в статью;
- в конце сборника генерируется статистический отчёт о количестве статей (если указана команда `\FINALSTAT`);

Поддержка рецензирования

Рецензенты могут вставлять замечания прямо в текст статьи.

► `\REVIEWERNOTE{текст замечания}`

Эта команда вставляет текст замечания жирным курсивом, выделяет его фигурными скобками и ставит на полях галочку с номером замечания. *{Вот так будут выглядеть замечания рецензента в тексте статьи.}* Рекомендуется начинать эту команду с отдельной строки, чтобы замечания были хорошо заметны в исходном тексте:

```
\REVIEWERNOTE{%
    Вот так будут выглядеть замечания рецензента в~тексте статьи.}
```

► `\NOREVIEWERNOTES`

Данная команда подавляет вывод всех замечаний рецензентов. Она может быть задана в преамбуле документа. В режиме чистовой печати сборника данная команда по умолчанию даётся в самом начале. Тем самым все замечания, которые остались в статьях, будут скрыты. В режиме черновой печати `draft` данная команда переопределяется на пустую. Тем самым все случайно оставленные команды `\NOREVIEWERNOTES` будут проигнорированы, и комментарии будут выведены во всех статьях.

► `\ACCEPTNOTE`, `\AMENDNOTE`, `\REJECTNOTE`

Эти команды выводят одно из трёх возможных решений по статье: принять к публикации, отправить на доработку, отказать в публикации. Они вставляются рецензентом или редактором в конец статьи перед `\end{document}`. Текст решения задаётся соответственно командами `\typeACCEPTNOTE`, `\typeAMENDNOTE`, `\typeREJECTNOTE`, которые можно переопределить (например, формулировку отказа можно изменить на более жёсткую, исключив возможность доработки). В стиле `mmro.sty` они определены следующим образом:

```
\newcommand\typeACCEPTNOTE{\par
    {\itshape\bfseries
        Статья принята к~публикации в~сборнике.}\par
}
\newcommand\typeAMENDNOTE{\par
    {\itshape\bfseries
        Статья принята к~публикации в~сборнике,
        однако желательна её доработка.}\par
}
\newcommand\typeREJECTNOTE{\par
    {\itshape\bfseries
        Статья может быть принята к~публикации в~сборнике
        только после существенной переработки.}\par
}
```

Важно помнить, что команды `\ACCEPTNOTE`, `\AMENDNOTE`, `\REJECTNOTE` только сообщают окончательное решение авторам статьи, но не влияют ни на включение статей в сборник, ни на подсчёт принятых и непринятых статей командой `\FINALSTAT` (всё это определяется исключительно символом γ команд `\paper` и `\FILTER`).

Все три команды `\ACCEPTNOTE`, `\AMENDNOTE`, `\REJECTNOTE` после вывода решения вызывают команду `\FINALREVIEWERNOTE`.

► `\FINALREVIEWERNOTE`

Эта команда подсчитывает общее число замечаний и выводит обращение к авторам. Использовать её при рецензировании не рекомендуется, так как она не выводит окончательное решение. Она вызывается автоматически командами `\ACCEPTNOTE`, `\AMENDNOTE`, `\REJECTNOTE`. Текст обращения генерируется командой `\typeFINALREVIEWERNOTE`, которую можно переопределить. В стиле `mmro.sty` она определена следующим образом:

```
\newcommand\typeFINALREVIEWERNOTE{%
  \par\noindent
  \begin{minipage}{\textwidth}\parindent=3.5ex
  \par\bigskip\hrule\nopagebreak\bigskip
  Всего сделано замечаний: \themmroReviewerNote.
  \par\medskip
  Уважаемые авторы!
  \par\medskip
  Ваша статья уже прошла стадию корректуры. Убедительная просьба
  сохранить редакторские правки при подготовке финальной версии
  статьи. Для этого, пожалуйста, исправляйте присланный Вам
  вариант статьи, а не Ваш исходный вариант.
  \par
  Рецензенты могли вставлять свои замечания прямо в текст статьи
  с помощью команды \texttt{\char"5C REVIEWERNOTE}. Удалять их
  из документа не нужно. При печати сборника они не будут видны.
  \par
  Чтобы распечатать статью, скрыв замечания рецензентов, включите
  в преамбулу команду \texttt{\char"5C NOREVIEWERNOTES}.
  \par\nopagebreak\medskip\hrule\nopagebreak\bigskip
  \end{minipage}
}
```

Данное обращение очень важно. Если автор будет править свой исходный текст, то возникнет две версии текста, которые могут отличаться огромным количеством мелочей. Их аккуратное слияние является, очевидно, изощрённой формой издевательства над корректорами. Как поступать в таких случаях — определяется политикой редколлегии. Однако

авторов обязательно надо предупреждать о том, что они должны править именно последнюю версию текста.

Настройка параметров

В этом разделе перечислены команды, которые можно переопределять для изменения внешнего вида заголовков статей, разделов и подразделов, списков литературы и оглавления. Для каждой команды приводится её определение в стиле `mmro.sty`.

► `\vskipBeforeTitle`, `\vskipAfterTitle`

Эти команды задают вертикальный промежуток до и после заголовка статьи. В стиле `mmro.sty` они определены следующим образом:

```
\setlength\vskipBeforeTitle{3ex}
\setlength\vskipAfterTitle{1ex}
```

► `\typeTitle`, `\typeAuthor`, `\typeOrganization`, `\typeEmail`

Эти команды задают внешний вид элементов заголовка статьи: название, автор или список авторов, организация и адрес соответственно.

```
\renewcommand\typeTitle[1]{\large\sffamily\bfseries #1}
\renewcommand\typeAuthor[1]{\itshape\bfseries #1}
\renewcommand\typeOrganization[1]{\small #1}
\renewcommand\typeEmail[1]{\ttfamily #1}
```

► `\typeTocAuthorTitle{автор(ы) статьи}{название статьи}`

Внешний вид заголовка статьи в оглавлении:

```
\renewcommand\typeTocAuthorTitle[2]
{\{\unstretchspaces\itshape #1\}\ #2}
```

Здесь команда `\unstretchspaces` преобразует неразрывные пробелы ~ в нерастяжимые \;, что улучшает вид списка авторов в тех случаях, когда он не помещается на одной строке.

► `\typeSection{заголовок}`

Внешний вид заголовков разделов `\section`, `\subsection` и `\subsubsection`. В стиле `mmro.sty` между этими разделами нет никакой разницы.

```
\renewcommand\typeSection[1]{%
\medskip\hangindent=3.5ex\hangafter=-3\noindent
{\normalfont\sffamily\bfseries #1}%
\par\nobreak\smallskip
}
```

► `\typeParagraph{заголовок}`

Внешний вид заголовков разделов `\paragraph` и `\subparagraph`. В стиле `mmro.sty` между этими разделами нет никакой разницы.

```
\renewcommand\typeParagraph[1]{%
  \smallskip{\normalfont\rmfamily\bfseries #1 }%
}
```

► `\typeChapter{код раздела}{расшифровка}{название раздела}`

Внешний вид заголовков разделов `\chapter`, стр. 2. Три аргумента этой команды имеют тот же смысл, что и в команде `\chapter`.

```
\renewcommand\typeChapter[3]{%
  \hrule\vskip1pt\hrule height2pt\vskip4ex\noindent
  {\normalfont\LARGE\sffamily\bfseries #3\raggedright\par}
  \vskip2ex\hrule height2pt\vskip1pt\hrule\vskip-1ex
  {\footnotesize\flushright Код раздела: #1 (#2)\par}
  \vskip6ex
}
```

► `\typeChapterCode {код раздела}`

Команда определяет внешний вид кода раздела, который выводится в колонтитулах:

```
\renewcommand\typeChapterCode[1]{(#1)}
```

Эту команду надо переопределять сразу после `\begin{papers}`. Если её переопределить на пустую, то коды разделов выводиться не будут.

► `\typeBibItem`

Внешний вид пункта библиографии. Обычно здесь устанавливается только размер шрифта.

```
\renewcommand\typeBibItem{\small\sloppy}
```

► `\BibAuthor {авторы}, \BibTitle {название}, \BibUrl {ссылка}`

Внешний вид элементов библиографии:

```
\renewcommand\BibAuthor#1{\emph{#1}}
\renewcommand\BibTitle#1{#1}
\renewcommand\BibUrl#1{{\small\url{#1}}}
\renewcommand\BibHttp#1{{\small\url{http://#1}}}
\renewcommand\BibFtp#1{{\small\url{ftp://#1}}}
```

Сборка алфавитного указателя авторов

Сборка алфавитного указателя авторов производится автоматически по всем командам `\author`, встретившимся в статьях. Если в команде указан список авторов через запятую, то каждый из них будет занесён в алфавитный указатель.

Чтобы алфавитный указатель был сформирован корректно, в команде `\author` не должно быть никаких посторонних символов. В частности, не допускается указывать с помощью звёздочек или индексов соответствие фамилий авторов с адресами и организациями, задаваемыми командами `\email` и `\organization`. Фамилии и инициалы должны задаваться только в формате `Фамилия~И.\,О.`

После компиляции сборника формируется файл-полуфабрикат с расширением `idx`. К сожалению, этот файл не может быть передан программе `makeindex` непосредственно, так как он содержит русские буквы в кодировке пакета `babel`. Сначала его необходимо преобразовать с помощью какой-нибудь программы автоматической замены символов, например, `sed`. К стилевому файлу `mmro.sty` прилагается файл `_mmro.sed`, содержащий правила перекодировки, и скрипт `_makeindex.bat` для корректной сборки индекса:

```
sed.exe -f _mmro.sed %1.idx > %1.id2
makeindex.exe -c -t %1.ilg -o %1.in2 -s _mmro.ist %1.id2
```

При вызове ему должно быть передано имя файла сборника без расширения. После отработки этого скрипта сборник должен быть откомпилирован ещё раз.

Алфавитный указатель вставляется в текст сборника командами

```
\newpage\input{\jobname.in2}
```

Генерация архивов

Во время компиляции сборника формируются `bat`-файлы, последующий запуск которых позволит собрать комплект архивов для рассылки авторам и рецензентам:

- `reviewers.bat` — скрипт для формирования архивов, рассылаемых рецензентам;
- `authors.bat` — скрипт для формирования архивов, рассылаемых авторам;
- `corr-authors.bat` — скрипт, аналогичный предыдущему, но архивируются только статьи, прошедшие корректуру и/или содержащие команду `\REVIEWERNOTE`.

► `\ZipAdd{имя архива без расширения}`

Данная команда определяет начало командной строки вызова архиватора. Предполагается, что если после строки, выведенной командой `ZipAdd`, будет следовать список имён файлов через пробелы, то получится корректная запись команды добавления указанных файлов в архив.

Команду `ZipAdd` можно переопределять. По умолчанию в `mmro.sty` она формирует архивы `rar`:

```
\newcommand\ZipAdd[1]{rar a #1.rar }
%\newcommand\ZipAdd[1]{zip -add #1.zip }
```

Отметим, что пробел после имени архива обязателен, без него имя архива сольётся с именем первого добавляемого файла, и при запуске `bat`-файла команда сработает некорректно.

Во время компиляции сборника архивы не формируются, так как это было бы слишком долго. Запуск сгенерированных `bat`-файлов должен производиться либо вручную, либо описанным ниже скриптом формирования сборника.

Генерация списка всех статей

Во время компиляции сборника формируются текстовый файл с именем `papers.tab`, содержащий таблицу статей. Каждая строка в этом файле соответствует ровной одной статье и состоит из следующих записей, разделяемых символом `|`:

- имя файла;
- код раздела, заданный командой `\chapter`;
- полное название статьи, заданное командой `\title`;
- полный список авторов через запятую, заданный командой `\author`;
- организация, заданная командой `\organization`;
- адрес или список адресов, заданные командой `\email`;
- список рецензентов и корректоров, заданный командой `\paper`;
- символ α (+ прошла корректуру, - не прошла);
- символ β (+ прошла рецензирование, - не прошла);
- символ γ (+ принята в печать, ? желательна доработка, ! необходима доработка, - не принята);
- количество замечаний `\REVIEWERNOTE`, вставленных рецензентами в текст статьи.

Данный текстовый файл может быть легко импортирован в любую базу данных или электронную таблицу для дальнейшей обработки; например, для организации рассылки или формирования HTML-страницы со списком принятых публикаций.

К сожалению, сформированный файл содержит русские буквы в кодировке пакета `babel`. Для его конвертирования в обычный текстовый файл можно воспользоваться любой программой автоматической замены символов. Например, `sed`:

```
sed.exe -f papers.sed papers.tab > papers_list.txt
```

На выходе имеем файл `papers_list.txt` в кодировке Windows 1251.

Файл `papers.sed` содержит правила перекодировки и идёт в комплекте со стилевым файлом `mmro.sty`.

Скрипт для формирования сборника

Скрипт `_make.bat` описывает всю последовательность действий, необходимых для вёрстки сборника.

Сначала удаляются все файлы, которые могли остаться от предыдущей сборки:

```
del *.aux *.log *.toc *.bak *.dvi *.idx *.ind *.ilg
```

Компилируется основной dvi-файл сборника:

```
TeXify MMR0-13.tex
```

Формируется алфавитный указатель авторов:

```
CALL _makeindex.bat MMR0-13
```

Все `tex`-файлы, находящиеся в директории, компилируются в `dvi`, затем в `pdf`:

```
FOR %%i IN (*.TeX) DO TeXify %%i
FOR %%i IN (*.dvi) DO DviPdfm -p a4 %%i
```

Во время компиляции были созданы три `bat`-файла для формирования комплекта архивов авторам и рецензентам. При необходимости некоторые из них могут быть вызваны:

```
rem CALL reviewers.bat
CALL authors.bat
rem CALL corr-authors.bat
```

И последнее: во все сформированные архивы должен быть добавлен стилевой файл `mmro.sty` самой последней версии. В данном случае предполагается, что он находится в текущей директории (иначе к его имени необходимо приписать полный путь):

```
FOR %%i IN (*.rar) DO rar a -ep %%i mmro.sty
```

Выполнение скрипта `_make.bat` для сборника из 230 статей общим объёмом около 800 страниц занимает около 5 минут.