

Пакет churchslavonic — верстка церковнославянских текстов в системе \LaTeX

Александр Андреев и Михаил Крутиков
версия v0.2.4

24 июня 2023 г.

Аннотация

Пакет churchslavonic позволяет верстать церковнославянские документы. В пакет включены шаблоны переносов, шрифты и набор необходимых макрокоманд.

Содержание

1	Как пользоваться пакетом	2
1.1	Опции color, gray (= grey) и bw	2
1.2	Знак подчеркивания	2
2	Числа	3
2.1	<code>\cuNum</code>	3
3	Даты	4
3.1	<code>\cuDate</code>	4
3.2	<code>\cuDateJulian</code>	5
3.3	<code>\cuDefineDateFormat</code>	5
3.4	<code>\cuUseDateFormat</code>	6
3.5	<code>\cuMonthName</code>	6
3.6	<code>\cuDayName</code>	6
3.7	<code>\cuDayNameAccusative</code>	6
3.8	<code>\cuToday</code>	6
3.9	<code>\cuTodayJulian</code>	7
3.10	<code>\cuAsJulian</code>	7
3.11	<code>\cuAsGregorian</code>	7

4 Киноварь	7
4.1 <code>\cuKinovar</code>	7
4.2 <code>\cuKinovarColor</code>	8
5 Разное	8
5.1 <code>\cuMarginMark</code> , <code>\cuMarginMarkSkip</code> и <code>\cuMarginMarkText</code> . . .	8
5.2 Надстрочный текст	9
5.3 Буквицы	10
6 Знаменная нотация	10
6.1 Набор Знаменной нотации	10
6.2 Управление внешним видом знаменных текстов	13

Введение

Церковнославянский язык (ISO 639-2 код `cu`) – древний литературный язык славянских народов, который теперь используется в основном как богослужебный язык в Русской Православной Церкви, других поместных православных церквях, а также в грекокатолических и старообрядческих общинах. Пакет `churchslavonic` позволяет верстать церковнославянские тексты в системе `TeX`. В пакет включены шаблоны переносов, шрифты и набор необходимых макрокоманд.

Данный пакет поддерживает только набор в стандарте Юникод. Тексты, набранные в устаревших кодировках HP и UCS, можно конвертировать в Юникод (точнее – в UTF-8) с помощью отдельных утилит, которые можно найти на сайте [Инициативной группы славянской информатики](#). Для того, чтобы использовать возможности этого пакета Вам понадобится система верстки `TeX`, поддерживающая Юникод, например, `XYTeX` или `LuaTeX`.

1. Как пользоваться пакетом

Добавьте следующие определения в преамбулу вашего документа:

```
\usepackage{polyglossia}
\setmainlanguage{churchslavonic}
\usepackage{churchslavonic}
```

Сразу станут доступны церковнославянские шаблоны переноса слов. После чего переключайтесь между языками стандартными средствами пакета `polyglossia`. Церковнославянские шрифты предоставлены в отдельном

пакете `fonts-churchslavonic`, который должен был установиться, когда вы установили этот пакет. Подробности о шрифтах смотрите в документации к пакету `fonts-churchslavonic`.

1.1. Опции `color`, `gray` (= `grey`) и `bw`

Опции позволяют отключать цвет раскраски текста в командах типа `\skipovar`. Полезны для создания версий для печати на устройствах, которые не поддерживают цветности.

- `color` эта опция используется по умолчанию и означает, что используется нормальный красный цвет.
- `gray` (`grey`) заменяет красный цвет серым — используйте эту опцию, если ваше устройство печати не поддерживает цветности, но может качественно передавать градации серого цвета.
- `bw` отключает всякую раскраску — используйте для создания черно-белой версии документа.

Пример:

```
\usepackage[gray]{churchslavonic}
```

1.2. Знак подчеркивания

Знак подчеркивания (`_`, U+005F Low LINE) в церковнославянском языке играет примерно ту же роль, что и дефис в русском. Пакет `churchslavonic` переопределяет знак подчеркивания так, что его можно набирать и в текстовой моде, например:

Послѣдованіе молебнаго пѣніа стѣмъ мѣкамъ кѣго вѣка,
въ Сѣнктѣ-Петерѣвѣржстѣй дѣхѣвнѣй академіи
начальствовавшимъ, учившимъ и учившимся

Внимание: если у вас установлена старая версия пакета `fontspec`, переопределенный знак подчеркивания нельзя использовать при задании имен и параметров шрифтов в командах `fontspec` типа `\setXXXfont` и `\newfontfamily`.

Обычно для церковнославянских шрифтов требуется установить подчеркивание как символ переноса, указав параметр: `NurphenChar=_`. Старые версии `fontspec` выдадут ошибку. Эта проблема присутствует в `TeX Live 2013` и в `fontspec v2.3c`.

Мы рекомендуем обновить систему T_EX до T_EX Live 2015 или свежее. Или же обновить только пакет fontspec до версии 2.4c или более новой.

Если же вы не можете установить обновления, то можно обойти проблему, указав символ переноса как код в шестнадцатиричном представлении: `HuphenChar="005F`. Другой подход — это декларировать все шрифты до загрузки пакета churchslavonic.

2. Числа

Способ записи чисел в церковнославянском языке (кириллическая цифирь) основывается на древнегреческом и в качестве цифр использует буквы. За подробностями отсылаем интересующегося читателя к соответствующей главе [1].

2.1. `\cuNum`

Команда печатает число кириллической цифирью. Она принимает единственный аргумент. Аргументом может быть текст или командная последовательность. Единственное условие — аргумент должен раскрыться в число.

<code>\cuNum{1}</code>	ѧ
<code>\cuNum{12}</code>	ѦѢ
<code>\cuNum{123}</code>	ѢѢѢ
<code>\cuNum{1234}</code>	ѠѦѢѢ
<code>\cuNum{10345}</code>	ѠѢѢѢѢ
<code>\cuNum{12345}</code>	ѦѢѢѢѢ
<code>\cuNum{123456}</code>	ѢѢѢѢѢѢ
<code>\cuNum{800456}</code>	ѠѢѢѢѢѢѢ
<code>\cuNum{1234567}</code>	ѠѢѢѢѢѢѢѢ
<code>\cuNum{1500567}</code>	ѠѢѢѢѢѢѢѢѢ
<code>\cuNum{12345678}</code>	ѦѢѢѢѢѢѢѢѢ
<code>\cuNum{123456789}</code>	ѢѢѢѢѢѢѢѢѢѢ

3. Даты

<code>\cuDate{2016-4-21}</code>	ѦѢ ѠѢѢѢѢѢѢ, ѠѢѢѢѢ ѢѢѢѢ
<code>\cuDateJulian{2016-4-21}</code>	ѢѢ ѠѢѢѢѢѢѢ, ѠѢѢѢѢ ѢѢѢѢ
<code>\cuDate{\cuToday}</code>	ѦѢ ѢѢѢѢѢѢ, ѠѢѢѢѢ ѢѢѢѢѢѢ

3.1. `\cuDate`

Команда форматирует дату (в соответствии с текущим форматом). Аргумент должен иметь вид YYYY-MM-DD. Результат может выглядеть примерно так: *кв̄ апріліа, лѣ҃та ,кѣ҃т*.

Отметим, что значение даты никоим образом не интерпретируется и не нормализуется. Поэтому можно вызывать макрокоманду и с невозможными датами, например 32 апреля — такая дата будет отформатирована как 32 апреля. Так что команду `\cuDate` можно использовать для набора фраз вроде «дата `\cuDate{2016-04-32}` не существует ни в одном календаре».

Однако, если вы используете свой формат даты и в этом формате задействованы символические переменные `\cuDOW` (день недели) или `\cuYEARAM` (год от сотворения мира), то значения этих переменных будут вычислены исходя из заданной даты — и дата интерпретируется как дата по григорианскому календарю («новому стилю»). В этом случае, невозможные даты будут нормализованы методом экстраполяции. Например, 32 апреля будет интерпретировано как 2 мая для целей определения дня недели и года от сотворения мира.

Если ваш формат использует день недели (`\cuDOW`) или год от сотворения мира (`\cuYEARAM`), и вы указываете дату по юлианскому календарю («старому стилю»), вы должны вместо `\cuDate` пользоваться макрокомандой `\cuDateJulian` для правильной обработки дня недели и года от сотворения мира.

Лучше всего всегда использовать `\cuDate` для форматирования дат по григорианскому календарю и всегда использовать `\cuDateJulian` для форматирования дат по юлианскому календарю — вне зависимости от значения текущего формата. Таким образом вы можете впоследствии изменить формат даты (например, добавив в него день недели) и результат останется правильным.

3.2. `\cuDateJulian`

Форматирует дату в соответствии с текущим форматом подобно команде `\cuDate`. Отличие заключается в том, что аргумент интерпретируется как дата по юлианскому, а не григорианскому календарю. Это имеет значение только если текущий формат даты использует символические имена `\cuDOW` (день недели) или `\cuYEARAM` (год от сотворения мира).

3.3. `\cuDefineDateFormat`

Команда определяет формат даты. Она никак не влияет на то, как `\cuDate` форматирует свой вывод (для этого используется `\cuUseDateFormat`).
Пример:

```
\cuDefineDateFormat{long}{%
  \cuDayName{\cuDOW},
  \cuNum{\cuDAY}\textunderscore gw~%
  \cuMonthName{\cuMONTH},~%
  лѣта ѿ сотворѣнїа міра~%
  \cuNum{\cuYEARAM}%
}
```

определяет новый формат с именем `long`. Если мы напечатаем дату этим форматом, то получим: пачѣокъ, къ_гв апрїлїа, лѣта ѿ сотворѣнїа міра ѡфкѣ.

При определении формата можно пользоваться следующими символическими именами:

- `\cuYEAR` — год (число, например 2016)
- `\cuYEARAM`¹ — год от сотворения мира по византийскому летоисчислению (число, например 7525).
- `\cuMONTH` — месяц (число от 1 до 12, где 1 означает «январь»)
- `\cuDAY` — день месяца
- `\cuDOW`¹ — день недели (число от 0 to 6, где 0 означает «воскресенье»)
- `\cuINDICTION` — индикт² (число от 1 до 15)

3.4. `\cuUseDateFormat`

Макрокоманда устанавливает имя текущего формата даты. Этот формат будет использоваться при последующих вызовах макрокоманд `\cuDate` и `\cuDateJulian`.

¹Если ваш формат пользуется этим значением, вы должны форматировать дату правильной макрокомандой: `\cuDate` для дат по григорианскому календарю и `\cuDateJulian` для дат по юлианскому календарю.

²См. <https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D0%B4%D0%B8%D0%BA%D1%82>

3.5. `\cuMonthName`

Команда раскрывает числовой аргумент в текстовое представление месяца. Команда обычно используется при задании формата даты. Например, формат с именем `default` определен так:

```
\cuDefineDateFormat{default}{%
  \cuNum{\cuDAY}~\cuMonthName{\cuMONTH},%
  ~\cuNum{\cuYEAR}%
}%
```

3.6. `\cuDayName`

Команда раскрывает числовой аргумент в текстовое представление дня недели в именительном падеже.

3.7. `\cuDayNameAccusative`

Команда раскрывает числовой аргумент в текстовое представление дня недели в винительном падеже.

3.8. `\cuToday`

Эта макрокоманда раскрывается в текущую дату в виде триплета `YYYY-MM-DD`. Дата выводится по григорианскому календарю («новому стилю»).

3.9. `\cuTodayJulian`

Эта макрокоманда раскрывается в текущую дату в виде триплета `YYYY-MM-DD`. Дата выводится по юлианскому календарю («старому стилю»).

Команда просто раскрывается в: `\cuAsJulian{\cuToday}`.

3.10. `\cuAsJulian`

Преобразует дату по григорианскому календарю в дату по юлианскому календарю. Дата задается и выводится как числовой триплет `YYYY-MM-DD`.

Команда полезна, когда одну и ту же дату надо написать и по григорианскому, и по юлианскому календарю.

3.11. `\cuAsGregorian`

Преобразует дату по юлианскому календарю в дату по григорианскому календарю. Дата задается и выводится как числовой триплет `YYYY-MM-DD`.

4. Киноварь

Печатные и рукописные книги на церковнославянском языке часто пользуются красным цветом («киноварью») для выделения структурных элементов текста. Обычно богослужебные указания («рубрики»), заголовки, замечания и пометы на полях печатаются красным цветом. Кроме того, первая буква каждого параграфа тоже часто выделяется красным цветом.

4.1. `\cuKinovar`

Команда печатает свой аргумент красным цветом. Например, если мы явно зададим текст аргумента, то получим ожидаемый результат:

<code>\cuKinovar{лѣкх:} гдѣ помѣлѣи.</code>	лѣкх: гдѣ помѣлѣи.
---	--------------------

Если же используется механизм `TeX` неявного задания аргумента, то команда будет применена к первой букве последующего текста. При этом команда “соберет” все необходимые выносные символы, так что все надстрочные знаки первой буквы будут также корректно раскрашены! Поэтому очень удобно использовать эту команду с неявным заданием аргумента для раскраски первой буквы параграфа.¹

<code>\cuKinovar Поимх гдѣи пѣснь нѣвѣю</code>	Поимх гдѣи пѣснь нѣвѣю
<code>\cuKinovar ѿкв чѣча на трѣскочѣ</code>	ѿкв чѣча на трѣскочѣ

4.2. `\cuKinovarColor`

Команда переключает текущий цвет на красный. Должна использоваться внутри группы, которая ограничит ее действие (если вы не хотите чтобы весь последующий текст был напечатан красным цветом).

Справка: Оттенок красного цвета, используемый командой `\cuKinovar` объявляется в пакете новым цветом с названием `kinovar`, и задан как (205, 8, 3) в RGB-цветовом пространстве, что равно `#CC0502` в шестнадцатирядном представлении.

¹Вы также можете поэкспериментировать со встроенной командой `TeX` `\everypar` с целью полностью автоматизировать раскраску первой буквы каждого параграфа. Однако успех сильно зависит от того, какой класс системы `TeX` используется и от того, какие пакеты подгружаются. Мы не нашли надежного способа заставить `\everypar` работать, и поэтому пакет `churchslavonic` не предлагает никаких макрокоманд для дальнейшей автоматизации этого процесса. На практике, часто проще и надежнее воспользоваться функцией текстового редактора Найти/Заменить для вставки макрокоманды `\cuKinovar` перед каждым параграфом.

5. Разное

5.1. `\cuMarginMark`, `\cuMarginMarkSkip` и `\cuMarginMarkText`

Команда `\cuMarginMark` используется для печати значков на полях, напротив строки, где вызвана эта команда. Например:

```
\cuMarginMark{б}\cuKinovar Блгоеловн дшше мо̀ гда̀ н не забывай̃  
всѣхъ возда́ннй̃ ѿг̀. \cuMarginMark{я}\cuKinovar Ѡчица̀юцаго  
всѣ̀ беззакѡнїа̀ твоѡ̀, ѡцѣла̀юцаго всѣ̀ недѣлїи твоѡ̀:  
\cuMarginMark{б}\cuKinovar Ѡзбавла̀юцаго ѡ̀ ѡстаѣнїа̀ живѡтцѣ̀ твоѡ̀,  
вѣнча̀юцаго тѣ̀ млчїю̀ н щедрѡтѣми: \cuMarginMark{я}\cuKinovar  
Ѡсполна̀юцаго во бл҃гїхъ желанїѐ твоѡ̀, ѡбновїтѣлѣ̀ ѡкѡ̀ Ѧрла̀  
ѡноеть твоѡ̀.
```

Получим такой результат:

Блгоеловн дшше мо̀ гда̀ н не забывай̃ всѣхъ возда́ннй̃ ѿ
ѿг̀. Ѡчица̀юцаго всѣ̀ беззакѡнїа̀ твоѡ̀, ѡцѣла̀юца- ѡ
го всѣ̀ недѣлїи твоѡ̀: Ѡзбавла̀юцаго ѡ̀ ѡстаѣнїа̀ живѡтцѣ̀ ѿ
твоѡ̀, вѣнча̀юцаго тѣ̀ млчїю̀ н щедрѡтѣми: Ѡсполна̀юцаго ѡ̀
во бл҃гїхъ желанїѐ твоѡ̀, ѡбновїтѣлѣ̀ ѡкѡ̀ Ѧрла̀ ѡноеть
твоѡ̀.

Помета помещается на «внешнее» поле, то есть справа от текста для нечетных страниц и слева от текста для четных.

Расстояние между пометой и текстом задается через значение `\cuMarginMarkSkip`. По умолчанию это:

```
\def\cuMarginMarkSkip{0.6em}
```

Шрифт и цвет пометы можно изменить, переопределив команду `\cuMarginMarkText`. Например, если Вы хотите чтобы все пометы печатались красным цветом, то достаточно поместить в преамбулу вашего документа следующее определение:

```
\def\cuMarginMarkText#1{\cuKinovar{#1}}
```

Если надо поменять размер, цвет, или шрифт для какой-то одной пометы, то проще всего это сделать при вызове `\cuMarginMark`:

```
\cuMarginMark{{\tiny *}}
```

5.2. Надстрочный текст

Макрокоманда `\cuSup` может быть использована для размещения произвольного надстрочного текста над текстом в строке, что бывает необходимо при издании текстов из славянских рукописей. Макрокоманда принимает два параметра: первый параметр – надстрочный текст, второй параметр – текст в строке. Например:

```
мол\cuSup{ва}{иѣт}
```

ѡА
МОЛИТ

Макрокоманда принимает необязательный параметр `raise`, который управляет вертикальным пространством между текстом в строке и надстрочным текстом. Форматирование строчного и надстрочного текстов также можно контролировать напрямую:

```
по\cuSup[raise=0.75ex]{\kern2em \cuKinovar{лагаѣть}}{\cuKinovar{лож}} наѣмь
```

ЛАГАѢТЬ
ПОЛОЖИ НАМЪ

5.3. Буквицы

Механизм, используемый командой `\cuKinovar` для сбора выносных символов (при неявном задании аргумента), может оказаться полезен в некоторых других случаях. Один из примеров — это “буквицы” в начале разделов. Для набора буквицы отлично работает стандартный пакет `lettrine`. Единственное неудобство — это необходимость задавать явно первую букву вместе со всеми ее выносными символами. Естественно желание упростить и автоматизировать эту работу задействовав тот же механизм, что и в `\cuKinovar`.

Вот как можно добиться этого:

```
\def\cu@lettrine{\lettrine[lines=3,findent=0pt,nindent=0pt]}  
\def\cuLettrine{\cu@tokenizeletter\cu@lettrine}  
\renewcommand{\LettrineFontHook}{\cuKinovarColor}
```

Поместите это определение команды `\cuLettrine` в преамбулу вашего документа (и не забудьте окружить его командами `\makeatletter` и `\makeatother`). После этого можно набирать буквицы так:

```
\cuLettrine Ѣже дѣла сила въ немоци совершѣется...
```


Хри-сто́съ ра-жда́-ет-са, сла́-ви-те: Хри-сто́съ съ нб́съ, сла́-щи-те:
 Хри-сто́съ на зе-мль, воз-но-си́-те-са. по́й-те Го́-спо-де-ви сла́-- зе-
 мля́,
 и́ ве-се́-лі-емь вос-по--йте лю́-ді-е, ю́-кв про-сла́-ви-са.}



Примечание 1: количество групп неvm должно равняться количеству слогов, в противном случае при компиляции макрокоманда выдаст ошибку, например: ! Too many kruk groups.

Примечание 2: Как в макрокоманде `\cuKruk`, так и в макрокоманде `\cuKrukPara`, символ `~` может быть использован, если какая-то неvm стоит самостоятельно, а не над слогом текста. В этом случае под неvmой будет нарисована горизонтальная черта. Горизонтальная черта также будет нарисована, если указан пустой слог. Если символ `~` введен в блок для неvmы (или блок для неvmы оставлен пустым), команды производят слог без неvmы над ним (в этом случае, место для неvmы остается пустым). Чтобы расположить неvmу над пустым текстовым блоком (без горизонтальной черты), можно ввести любую другую макрокоманду, раскрывающуюся в пробел (например, `\thinspace`). Следующие примеры иллюстрируют эти возможности:

<code>\cuKruk{~}{}</code>	✓
<code>\cuKruk{~}{~}</code>	—
<code>\cuKruk{~}{~}</code>	✓
<code>\cuKruk{~}{}</code>	—
<code>\cuKruk{~}{\thinspace}</code>	✓
<code>\cuKruk{~}{text}</code>	text


Примечание 3: шрифты для Знаменной нотации обычно предоставляют информацию о цветах некоторых глифов (например, киноварных помет) в таблицах COLR и CPAL. Однако технология COLR и CPAL на данный момент не поддерживается в $\text{Xe}_\text{L}\text{a}_\text{T}_\text{E}_\text{X}$ и $\text{Lua}\text{L}_\text{a}_\text{T}_\text{E}_\text{X}$. Если пакет `churchslavonic` загружен с опцией `autocolormarks`, команды `\cuKruk` и `\cuKrukPara` автоматически раскрашивают киноварные пометы, внутренне вызывая для этого команду `\cuKinovar`. К тому же, если пакет `churchslavonic` также загружен с

опциями `gray` или `bw`, киноварные пометы будут набраны в оттенках серого или черным цветом, соответственно.

Ограничения: автоматическая раскраска киноварных помет в Xe_{La}TeX нарушает работу правил позиционирования глифов в таблицах OpenType. Попытка загрузить пакет `churchslavonic` с опцией `autocolormarks` когда используется Xe_{La}TeX выдаст предупреждение. Правильного позиционирования раскрашенных помет можно достичь в Lua_{La}TeX. Когда для верстки документа используется Lua_{La}TeX, пакет `churchslavonic` загружается с опцией `autocolormarks` по умолчанию. В этом случае автоматическое раскрашивание помет можно отключить, загрузив пакет `churchslavonic` с опцией `noautocolormarks`.

Макрокоманды `\cuKruk` могут вкладываться друг в друга, что позволяет набирать текст с «подобными пометами»:

```
\cuKruk{\cuKruk{\cuKinovar{\tiny ъ}}}  
\Large ъ}{ла}
```



6.2. Управление внешним видом знаменных текстов

Значение некоторых параметров может быть изменено, чтобы повлиять на позиционирование и внешний вид текстов со Знаменной нотацией:

<code>krukFont</code>	Указывает шрифт, используемый для отображения неvm
<code>sylSpace</code>	Контролирует количество пустого пространства вокруг слога (по умолчанию: 0.2em)
<code>topMargin</code>	Контролирует количество пустого пространства (поле) над неvмами (по умолчанию: 0.3em)
<code>krukRaise</code>	Контролирует длину разрыва между текстом и неvмами (по умолчанию: 1em)
<code>sylRuleHeight</code>	Указывает толщину горизонтальной черты, используемой, когда под неvmой нет слога (по умолчанию: 0.08em)

Эти параметры могут быть заданы как опции к макрокомандам `\cuKruk` и `\cuKrukPara`, разделенные запятыми. К примеру, повлияем на разрывы и поля в приведенном выше примере:

```
\cuKrukPara[krukRaise=1.5em,topMargin=0.6em,sylRuleHeight=0.02em]{ъ ъ ... \\  
Хри-стоꙋсъ ... }
```

Хри стѠсѧ ра ждѧ ет ѡд, ѡла вн те: Хри стѠсѧ ѡз нѣсѧ, ѡрѧ цн те:
Ѡ хри стѠсѧ на зѣ млѧ, воз но сі те ѡд. поѡ те Го спо де вн вѡд – зѣ
млѧ, ѡ вѣ сѣ лї емѧ вас по – ѡте лю дї е, ѡ кѡ про ѡла вн ѡд.

Список литературы

- [1] Aleksandr Andreev, Yuri Shardt, and Nikita Simmons. *Church Slavonic Typography in Unicode*, Unicode Technical Note 41. 2015. <http://www.unicode.org/notes/tn41/>