

A Package for Church Slavonic Typesetting

Andrey Slepukhin

Introduction

The multilingual ability of \TeX is one of its most important properties. Due to \TeX it has become possible to produce high-quality books in many different languages (sometimes with very exotic grammatical rules). During the past 16 years, \TeX has become a real polyglot and it seems that it doesn't want to stop evolving. In this paper one more, maybe rather exotic, example of practical usage of \TeX is considered, along with many ideas and solutions which result from five years of experience with \TeX .

General solutions

What does a language-specific package have to look like from the point of view of a computer typesetting system? It must include at least the following components:

- quality fonts;
- tools for simplifying the text formatting;
- hyphenation table;
- punctuation or some other poligraphic rule description.

These requirements form the basis of Slav \TeX development. The first two items have been realized satisfactorily. As to the realization of the third one — it depends on the volume of the dictionary, which is not sufficiently complete yet. The fourth item is absent because Church Slavonic has no precise rules for punctuation and other similar objects.

Fonts

Designing quality fonts is, in general, a very hard task; moreover, the author's knowledge on the subject at the beginning of this work was minimal. So, the designing of the base version of fonts took more than half a year, and various improvements are still under development. The work has been complicated by the fact that a large number of symbols (there are already 44 purely alphabetic characters) in the Church Slavonic alphabet must be included. Also the glyphs for symbols have very few shared elements. The typeface, which was in wide use at the beginning of the twentieth century, was taken as a model for the fonts created here. The following technology was applied to develop the fonts: the symbols were magnified and separate elements extracted, then base and control points of outline curves were placed manually and the METAFONT macros were designed; the symbols obtained were finally improved using the METAFONT graphic output. The

current font version consists of 148 symbols including some old Slavonic letters which are no longer in use.

The diacritic problem

In developing Slav \TeX , the main problem was that every word in a Church Slavonic text has at least one diacritic. None of the computer typesetting systems known to the author (except \TeX , of course), contains any convenient tools for typesetting a text with accents. \TeX uses the `\accent` macro for this purpose, but this macro seems to be designed for rather infrequent usage, because it gives the following undesirable effects:

- the kern between accented and previous symbols disappears;
- \TeX doesn't allow any hyphens in the remainder of the word after an accented symbol and can make invalid hyphens in the initial part of the word.

These effects arise because \TeX uses explicit kerning while expanding the `\accent` macro. So, it seems that the best solution for the diacritic problem (realized for many European languages, for example) is a method whereby a letter together with an accent is represented by a single character in the font. However, in the case of Church Slavonic, this solution cannot be applied in this form because there are too many possible 'letter-accent' pairs, and the limit of 256 symbols would soon be exhausted.

It would be wonderful if the following idea worked: the various 'letter-accent' combinations would be placed in a font at identical positions modulo 256 (i.e., 256 positions apart), so that their metrics would coincide. Unfortunately, it is not possible to force \TeX to put a symbol with character code greater than 255 into the `dvi` file. Such a restriction is quite unexpected since the `dvi` file format supports the use of symbols with character codes up to $2^{32} - 1$. One other well-known method to deal with the accents is their realization as strongly shifted left characters of zero width. Such an option is unsatisfactory too because it does not solve the kerning problem while significantly complicating construction of the hyphenation table.

To solve the accent problem we need to understand where diacritics are placed in Church Slavonic. Some can be placed only over the first letter in the word, and some can be placed only over the last letter. These two cases are realized by the special macros `\fcaccent` and `\lcaccent`. The second macro can be written in a very simple way because it only needs to locate the accent with

the help of kerns. The macro `\fcaccent` has a `\nobreak\hskip0pt` construction in addition, which enables the hyphenation of the word after the diacritic.

As for the accents in the middle of a word, some of them are realized together with the corresponding letters, and other represent symbols, in general used to abbreviate certain words (in Church Slavonic they are called *titlo*). The words containing these symbols cannot, as a rule, be hyphenated, but it was possible to write a special macro, placing an accent and preserving the kerns both before and after the symbol. It is a quite sophisticated macro which uses such \TeX commands as `\futurelet`.

To make inputting text easier, the symbols `'`, `"`, `‘`, `~`, `_`, `|` and `<` are made active and are expanded to the corresponding macros. The selection of Church Slavonic mode is realized by the `\beginslav` macro, and the return to normal mode is realized by the `\endslav` macro.

Separating colors

Another problem that had to be solved during package development was that of color separation. Almost all Church Slavonic texts are two-colored. Unfortunately, using colors via PostScript was impractical, because a PostScript-printer is a rarity in Russia. So, to make color separations and to obtain separate slides for each color, the $\text{SL}\TeX$ idea of using *invisible* fonts was applied. However, kerning problems make the use of $\text{SL}\TeX$ impossible. Indeed, having a word with a first letter emphasized by the use of another color (in Church Slavonic texts this occurs very often), $\text{SL}\TeX$ loses the required kern between the first letter and the remainder of the word when switching to the other font. So, for such cases we need special macros. To implement the color separation a special font selection scheme was designed, somewhat similar to NFSS. After including the font description file and appropriate macros, the user can declare the use of any color via the macro `\newcolor(<color>)`. This macro gives rise to the macros `\<color>g{<any text>}` and `\<color>`. The first of these switches the color, preserving kerning, and the second switches it without preserving any implicit kern (\TeX interprets this macro in the simplest way, so its usage makes sense). Now, typing `\showcolor(<color>)` or `\hidecolor(<color>)` in the input file, we can make any selection by a specified color visible or invisible in the output. The text before the first usage of `\<color>g{<any text>}` or `\<color>` will always be visible.

Numbering

In Church Slavonic, literal numeration is accepted, which can be described by the following algorithm:

Given an integer $n \geq 0$, let $S(n)$ be its representation in Church Slavonic. Consider the following table:

| n | $S(n)$ | n | $S(n)$ | n | $S(n)$ |
|---|--------|----|--------|-----|--------|
| 1 | ѧ | 10 | Ѧ | 100 | Ѱ |
| 2 | Ѩ | 20 | ѧ | 200 | ѱ |
| 3 | ѩ | 30 | Ѩ | 300 | Ѳ |
| 4 | Ѫ | 40 | ѩ | 400 | ѳ |
| 5 | ѫ | 50 | Ѫ | 500 | Ѵ |
| 6 | Ѭ | 60 | ѫ | 600 | Ѷ |
| 7 | ѭ | 70 | Ѭ | 700 | ѷ |
| 8 | Ѯ | 80 | ѭ | 800 | Ѹ |
| 9 | ѯ | 90 | Ѯ | 900 | ѹ |

The representation of zero is absent in Church Slavonic, but let it be empty for convenience.

If $10 \leq n < 20$, then $S(n) = S(n \bmod 10)S(10)$. If $20 \leq n < 100$, then $S(n) = S(n - (n \bmod 10))S(n \bmod 10)$. If $100 \leq n < 1000$, then $S(n) = S(n - (n \bmod 100))S(n \bmod 100)$. If $1000 \leq n < 10000$, then $S(n) = S(n - (n \bmod 1000))S(n \bmod 1000)$.

There are disagreements about the representation of numbers greater than 9999. In the $\text{Slav}\TeX$ package, we use a modern option, where the rule $S(n) = S(n - (n \bmod 1000))S(n \bmod 1000)$ is true for all numbers ≥ 1000 . The macro `\slnum(<number>)` automatically generates the number representation in Church Slavonic. For example, `\slnum(1995)` gives `ѰѦѩѧѩ`. One has to be careful since this macro is valid only inside Church Slavonic mode.

\TeX without encoding

During the development of $\text{Slav}\TeX$, an idea appeared which solves the compatibility problem when transferring any package to another platform. This problem is especially acute in Russia because Russian letter encodings on different platforms do not coincide. A version of \TeX Cyrillization made by CyrTUG is specific for PC-compatible computers under MS-DOS. This results in a justified unhappiness amongst the many UNIX users in big research institutes which need \TeX most of all.

The procedure to easily transfer any \TeX package to different platforms is given below:

- The encoding table containing a map between character codes and their symbolic names (for example, like PostScript names) must be defined for each specific platform and font family.

- A utility (which can even be written in \TeX !) must be created to generate two files from the original encoding table—a \TeX encoding table and a METAFONT encoding table.
- A set of METAFONT macros must be added to redefine the `beginchar` macro; it must allow the use of symbolic names instead of character codes by declaring `usenames:=1` or whatever like this.
- A hyphenation table must be written, using symbolic names; when generating the base file, \TeX should first read the encoding, then it should convert the original hyphenation table into a temporary file using the current encoding and then it should read the file obtained.
- `\catcode`, `\lccode` and `\uccode` should be defined using symbolic names.

A variant of this idea is implemented in the latest version of the package presented and is now being tested. It is hoped that new Cyrillization versions from CyrTUG will be written in the form described above. It would facilitate the work of the many \TeX users in Russia and of people who need to typeset Russian (or other Cyrillic) texts.

Type 1 from METAFONT?

One more idea, implemented as part of the Slav \TeX project, was inspired by an article by Jackowski and Ryćko [2]. Moreover, the arrival of a PostScript-printer on the author's desk helped stimulate its realization. There exists a set of METAFONT macros with which one can obtain a text representation of Type 1 fonts from METAFONT sources and from that, a downloadable font by L. Hetherington's Type 1 utilities. This macro package was initially designed to solve the specific problem of representing Church Slavonic in Type 1 format, but the conversion of Computer Modern fonts (and others) is also possible. This work deserves separate treatment and is not described in this paper.

Problems and plans

The most important problem today is adapting Slav \TeX to $\text{\LaTeX} 2_{\epsilon}$, or rather, its realization as a $\text{\LaTeX} 2_{\epsilon}$ package. The author also plans to develop a package for typesetting music in non-linear notation (so-called *krjuki*), in use before the eighteenth century. The following problems associated with Church Slavonic typesetting also should be noted: designing a font of initial caps and a special font for headings. In this font different combinations of letters must have specific glyphs (this task seems to be rather gigantic, because such a font would have a monstrous number of symbols and ligatures).

It would be nice if others shared the author's interest in the problem of Church Slavonic and ancient texts. Maybe at some future date, a multilingual edition of the Bible (in Church Slavonic, Greek, Latin, Hebrew ... what else?) produced with \TeX could come into existence.

Examples

A simple example of a Church Slavonic text:

```
Г|сди <i_исе х|срт'е, с_не б_жій,  
пом'илуї м'я гр'ешнаго
```

and the result of its compilation:

```
Гдн іісе хртѣ, сіе вѣїї, помілдї мѧ грѣшнаго
```

An example of color separation: a sequence of macros

```
\beginslav\family(slav)\size(12)%  
\black%  
\def\pray{%  
\redg Г|с{ди} <i_исе х|срт'е, с_не  
б_жій, пом'илуї м'я гр'ешнаго  
}%  
\black%  
\hidecolor(black)%  
\showcolor(red)%  
\par\noindent\pray  
\hidecolor(red)%  
\showcolor(black)%  
\par\noindent\pray  
\showcolor(red)%  
\par\noindent\pray  
\endslav
```

gives the result

```
Г  
Гдн іісе хртѣ, сіе вѣїї, помілдї мѧ грѣшнаго  
Гдн іісе хртѣ, сіе вѣїї, помілдї мѧ грѣшнаго
```

This example shows that accents can be placed over a group of characters, and not only over a single character.

References

- [1] Alipiy, Ieromonakh (Gamanovich). *Grammatika tserkovno-slavjanskogo jazyka*. Moscow: Palomnik, 1991.
- [2] Jackowski, Bogusław, and Marek Ryćko. "Labyrinth of METAFONT paths in outline", *Proceedings of the Eighth European \TeX Conference* (Sept. 26–30, 1994, Gdańsk, Poland), 18–32.
- [3] Donald E. Knuth. *The \TeX book*. Addison Wesley, Reading, MA, 1990.
- [4] Donald E. Knuth. *The METAFONTbook*. Addison Wesley, Reading, MA, 1990.

- [5] *Slovar' russkogo jazyka XI–XVII vv.* Moscow: Nauka, 1975.

◇ Andrey Slepukhin
Lavra, Sergiev Posad, Russia
Email: pooh@shade.msu.ru

Пакет для набора церковно-славянских текстов

Андрей Слепухин

Введение

Одним из самых важных свойств \TeX 'а безусловно является его многоязычность. Благодаря \TeX 'у стало возможным издание книг на самых разных языках (порой с весьма прихотливыми грамматическими правилами) с высочайшим полиграфическим качеством. За 10 с лишним лет своего существования \TeX стал настоящим полиглотом и, похоже, не собирается останавливаться на достигнутом. В этой статье рассматривается еще один, быть может, довольно экзотический пример применения многоязычности \TeX 'а на практике, а также многие идеи и решения, возникшие в результате 5-летнего опыта работы с \TeX 'ом.

Общий обзор

Что с точки зрения компьютерной издательской системы должен представлять собой пакет для работы с каким-либо языком? Он должен содержать по крайней мере следующие компоненты:

- качественные шрифты;
- средства облегчения набора текста;
- таблицу переносов;
- описание полиграфических правил, используемых в данном языке;

Эти требования и стали основой при разработке пакета $\text{\textbackslash}\text{sl}\text{\textbackslash}\text{TeX}$. Первые два пункта удалось реализовать вполне удовлетворительно, для качественной реализации третьего пока не хватает необходимого объема словаря, а четвертый практически отсутствует, так как в церковно-славянском языке каких-либо полиграфических правил просто нет.

Шрифты

Разработка шрифтов вообще является очень трудоемким занятием, к тому же познания автора в этом вопросе к началу работы были минимальными. Поэтому на разработку базового варианта шрифтов ушло больше полугода, а внесение различных улучшений продолжается до сих пор.

Среди факторов, усложнивших работу, следует отметить то, что церковно-славянский алфавит содержит большое количество символов (только букв—44), и начертания этих символов имеют мало похожих элементов. В качестве образца был взят шрифт, получивший широкое распространение в начале XX века. Технология разработки шрифта была следующая: символы увеличивались, разбивались на отдельные элементы, затем вручную приблизительно проставлялись опорные точки и эти элементы описывались с помощью макрокоманд $\text{\textbackslash}\text{METAFONT}$ 'а; полученные символы далее доводились с использованием графического вывода $\text{\textbackslash}\text{METAFONT}$ 'а. В настоящий момент шрифт содержит 148 символа, включая некоторые буквы старо-славянского языка, вышедшие из употребления.

Проблема диакритических знаков

Основная проблема, возникшая при разработке пакета $\text{\textbackslash}\text{sl}\text{\textbackslash}\text{TeX}$, была связана с тем, что в церковно-славянских текстах каждое слово имеет по крайней мере один диакритический знак. Ни одна из известных автору систем компьютерного набора не имеет удовлетворительных средств для набора текстов с диакритическими знаками. Лучший из известных способ предлагает \TeX . Для этой цели он использует макрокоманду $\text{\textbackslash}\text{accent}$, но она, видимо, была рассчитана лишь на достаточно редкое применение, потому что использование этой макрокоманды вызывает два нежелательных эффекта:

- исчезает kern между акцентируемым символом и предыдущим;
- конец слова, начиная с акцентируемого символа, не переносится вообще, а начало может переноситься неправильно;

Эти эффекты возникают из-за того, что \TeX , при реализации макрокоманды $\text{\textbackslash}\text{accent}$ использует явный kern. Поэтому, вероятно лучшим решением (что и реализовано, например, для некоторых европейских языков) является реализация буквы вместе с акцентом в виде отдельного символа в шрифте. Однако, в случае с церковно-славянским языком такое решение в чистом виде не проходит, поскольку возможных комбинаций буква—диакритический знак так много, что 256 символов в шрифте просто не хватит! Наверное, очень хорошим решением проблемы было бы следующее: различные пары буква—диакритический знак имеют в шрифте позиции совпадающие по модулю 256, и их метрики в TFM-файле совпадают. К сожалению, никаким

образом от TeX'a нельзя добиться, чтобы он в DVI-файл вывел символ с кодом большим, чем 256. Такое ограничение тем более непонятно, поскольку формат DVI-файла поддерживает использование символов с кодами до $2^{32} - 1$. Еще один известный способ борьбы с диакритическими знаками—делать их в виде символов с нулевой шириной и сильно смещенных влево. Такой вариант тоже является неудовлетворительным, поскольку проблему с кернингом он не решает и, кроме того, построение таблицы переносов в этом случае было бы делом весьма затруднительным.

Чтобы решить проблему, пришлось понять, по какому принципу расставляются диакритические знаки в церковно-славянском языке. Оказалось, что некоторые из них встречаются только над первой буквой слова, а некоторые—только над последней. Эти два случая реализуют специальные макрокоманды `\fcaccent` и `\lcaccent`. Написание последней не представляет особого труда, так как она лишь размещает с помощью кернов диакритический знак. В макрокоманде `\fcaccent` кроме этого, используется конструкция `\nobreak\hskipOpt`, которая после выравнивания диакритического знака с помощью кернов разрешает перенос слова.

Что касается тех диакритических знаков, которые встречаются в середине слова, то часть из них реализована вместе с соответствующими буквами в виде отдельных символов, а часть представляет собой знаки, применяемые, в основном, для сокращения написания некоторых слов (в церковно-славянском языке они носят название "титло"). Слова, содержащие эти знаки, как правило, не переносятся, поэтому стало возможным написать макрокоманду, которая ставит над символом акцент, сохраняя kern как до, так и после этого символа. Это довольно хитрая макрокоманда, в частности, для ее написания пришлось применить такое средство, как `\futurelet`.

Для удобства набора текста символы `'`, `"`, `'`, `~`, `_`, `|` и `<` сделаны активными и раскрываются в соответствующие макрокоманды. Переход в церковно-славянский режим осуществляется макрокомандой `\beginslav`, а выход—макрокомандой `\endslav`.

Разделение цветов

Другая проблема, с которой пришлось столкнуться при написании пакета—это разделение цветов. Практически все церковно-славянские тексты являются двуцветными. К сожалению, от реализации разделения цветов через Post-

Script пришлось отказаться, так как в России PostScript-принтер все еще является большой редкостью. Поэтому чтобы реализовать разделение цветов и получить отдельные слайды для каждого текста, была использована идея SLiTeX'a об использовании "невидимых" шрифтов. Однако, из-за проблем с кернингом, использовать SLiTeX в чистом виде не удастся. В самом деле, если у нас есть слово, первая буква которого выделена другим цветом (а такое в церковно-славянских текстах случается очень часто), то kern между выделенной буквой и остальным словом пропадает в связи с переключением на другой шрифт. Поэтому, для такого рода выделений приходится использовать специальные макрокоманды. Чтобы реализовать разделение цветов, была реализована специальная система подключения шрифтов, немного похожая на NFSS. После этого пользователь может описать используемые цвета макрокомандами `\newcolor(<цвет>)`. Эта макрокоманда порождает макрокоманды `\<цвет>g{<текст>}` и `\<цвет>`. Первая из них переключает цвет с сохранением кернинга, вторая—без сохранения кернинга (она более просто интерпретируется TeX'ом, поэтому ее использование имеет смысл). Теперь, если во входном файле указать `\showcolor(<цвет>)`, то выделения данным цветом будут видны при печати. Текст, расположенный до первой макрокоманды `\<цвет>` или `\<цвет>g` будет видимым независимо от макрокоманд `\showcolor`.

Нумерация

В церковно-славянском языке принята буквенная нумерация, описываемая следующим алгоритмом:

Пусть n —целое число, $n \geq 0$, а $S(n)$ —его представление в церковно-славянском языке. Для начала введем таблицу:

| n | S(n) | n | S(n) | n | S(n) |
|---|------|----|------|-----|------|
| 1 | ā | 10 | ī | 100 | ř |
| 2 | ĕ | 20 | ķ | 200 | č |
| 3 | ř | 30 | ā | 300 | č |
| 4 | ā | 40 | ā | 400 | ŷ |
| 5 | ē | 50 | ī | 500 | ϕ |
| 6 | š | 60 | ž | 600 | X |
| 7 | ž | 70 | ō | 700 | ψ |
| 8 | ī | 80 | ī | 800 | ŵ |
| 9 | ā | 90 | č | 900 | ц |

Представление числа 0 в церковно-славянском языке отсутствует, но для удобства будем считать его пустым.

Если $10 \leq n < 20$, то $S(n) = S(n \bmod 10)S(10)$. Если $20 \leq n < 100$, то $S(n) = S(n - (n \bmod 10))S(n \bmod 10)$. Если $100 \leq n < 1000$, то $S(n) = S(n - (n \bmod 100))S(n \bmod 100)$. Если $1000 \leq n < 10000$, то $S(n) = S(n - (n \bmod 1000))S(n \bmod 1000)$.

Относительно представления чисел, больших, чем 9999, существуют разногласия, в пакете \LaTeX реализован современный вариант, где правило $S(n) = S(n - (n \bmod 1000))S(n \bmod 1000)$ распространяется на все числа ≥ 1000 . Автоматическую генерацию записи чисел в церковно-славянском языке реализует макрокоманда $\text{\slnum}<\text{число}>$. Например, $\text{\slnum}(1995)$ даст *дцѣе*. Будьте внимательны: макрокоманда \slnum действует только в церковно-славянском режиме.

TeX без кодировок

В процессе работы над \LaTeX 'ом появилась идея, которая позволяет решить проблемы совместимости при переносе какого-либо пакета на другую платформу. Эта проблема особенно актуальна в условиях России, поскольку на разных платформах кодировка русских букв различна. Версия кириллизации \TeX 'а, распространяемая CugTUG, рассчитана в основном на пользователей IBM PC-совместимых компьютеров, работающих под MS DOS. Это, в частности, вызывает справедливое недовольство многочисленных пользователей UNIX'а, среди которых—крупные научные институты, которые в \TeX 'е нуждаются больше всего.

Идея легкого переноса на другую платформу заключается в следующем:

- Вводится таблица кодировки для конкретной платформы и конкретного семейства шрифтов, содержащая соотношение между символическими кодами и их именами (например, используемыми в PostScript'e);
- Определяется утилита (она может быть написана даже на \TeX 'e!), которая из таблицы кодировки генерирует два файла: первый—таблица кодировки для METAFONT'а, второй—для \TeX 'а
- Добавляется набор макрокоманд для METAFONT'а, который переопределяет макрокоманду \beginchar так, что при установке \usenames:=1 в ней могут быть использованы символические имена;
- Таблица переносов описывается с использованием символических имен; при генерации формата \TeX читает таблицу кодировки, за-

тем конвертирует таблицу переносов в промежуточный файл с использованием текущей кодировки и читает полученный файл;

- \catcode , \lccode и \uccode описываются также с использованием символических имен;

Вариант этой идеи реализован в последней версии представленного пакета и находится в процессе тестирования. Хочется надеяться, что аналогичным образом будут оформлены и последующие версии кириллизации \TeX 'а, распространяемые CugTUG, во всяком случае это облегчило бы работу для многих пользователей \TeX 'а в России, а также для всех тех, кому необходимо работать с русскими текстами.

Type 1 из METAFONT'а?

Еще одна идея, реализованная при разработке \LaTeX 'а была навеяна статьей [3], а также появлением на столе автора PostScript-принтера. В настоящее время имеется набор макроопределений для METAFONT'а, с помощью которого можно получить шрифт в формате Type 1 в текстовом виде, а затем, используя Type 1 утилиты, написанные L. Hetherington'ом,—в виде загружаемых шрифтов. С помощью этого набора макрокоманд можно получить представление в виде Type 1 и шрифтов Computer Modern. Эта работа заслуживает отдельного рассмотрения и, поэтому в данной статье не описывается.

Проблемы и планы

Самой важной из существующих на сегодняшний день проблем остается подключение пакета \LaTeX к \LaTeX 2_ϵ , точнее его оформление как \LaTeX 2_ϵ -пакета. В будущем автор также собирается заняться разработкой пакета для записи музыки в безлинейной нотации (так называемые "крюки"), использовавшейся в России вплоть до XVII века. В числе проблем, связанных с церковно-славянским языком, можно упомянуть разработку шрифтов для буквиц, а также—специального шрифта для заголовков, в котором различные буквосочетания имеют различные начертания (эта задача кажется весьма фантастичной, так как подобный шрифт должен иметь очень много символов и огромное число лигатур).

Хочется надеяться, что кто-нибудь разделит интерес автора к проблеме церковно-славянских и древних текстов. Может быть, когда-нибудь появится многоязычное издание Библии (на церковно-славянском, греческом, латинском, еврейском ... на каком еще?), выполненное с помощью \TeX 'а.

Примеры

Простой пример набора текста на церковно-славянском языке:

```
Г|сди <i_исе х|срт'е, с_не б_жій,
пом'илуй м'я гр'эшнаго
```

и результат его компиляции:

```
Гдн іісе хрт'е, сіе вжій, помілуй мѧ грѣшнаго
```

Пример цветodelения: последовательность команд

```
\beginslav\family(slav)\size(12)%
\def\pray{%
\redg Г|с{ди} <i_исе х|срт'е, с_не
б_жій, пом'илуй м'я гр'эшнаго
}%
\black%
\hidecolor(black)%
\showcolor(red)%
\par\noindent\pray
\hidecolor(red)%
\showcolor(black)%
\par\noindent\pray
\showcolor(red)%
\par\noindent\pray
\endslav
```

приводит к следующему результату:

```
Г
Гдн іісе хрт'е, сіе вжій, помілуй мѧ грѣшнаго
Гдн іісе хрт'е, сіе вжій, помілуй мѧ грѣшнаго
```

В этом примере хорошо видно, что акцент можно ставить не только над одним символом, но и над сочетанием символов.

Список литературы

- [1] Donald E. Knuth. *The T_EXbook*. Addison Wesley, Reading, MA, 1990.
- [2] Donald E. Knuth. *The METAFONTbook*. Addison Wesley, Reading, MA, 1990.
- [3] Bogusław Jackowski, Marek Ryćko. Labyrinth of METAFONT paths in outline. *EuroT_EX Proceedings, 1994: 18–32*.
- [4] Иеромонах Алипий (Гаманович). *Грамматика церковно-славянского языка*. Паломник, Москва, 1991.
- [5] *Словарь русского языка XI–XVII вв.* Наука, Москва, 1975.

◇ Андрей Слепухин
Лавра, Сергиев Посад, Россия
Email: pooh@shade.msu.ru

Production Notes on the Russian Papers

Michel Goossens

The two previous articles both have their first part in English, then an equivalent text in Russian, and finally a part with examples in both languages. Apart from these similarities the Russian text was coded in completely different ways for both articles.

The Lapko-Makhovaya article uses the KOI8 encoding popular in the Unix community in Russia. The KOI8 input was matched by hyphenation patterns and a font layouts using the same encoding.

On the other hand the Slepukhin article was coded in the Alternativniy encoding popular on PC machines, and the normal Russian and SlavT_EX fonts matched that encoding, as did the Russian hyphenation patterns needed to run that paper.

Both Russian encodings are 8-bits wide and coincide with ASCII in positions 0 to 127, but they place the Cyrillic characters in completely different locations between 128 and 255¹.

To run both articles two different formats had to be generated. The first contained the English and Russian hyphenation patterns using the KOI8 encoding, the second one English and Russian in the Alternativniy encoding. Also different style files and font encodings were necessary. Both font instances are based on the LH Cyrillic font family developed by Olga Lapko and her colleagues of CyrTUG [1], only the encoding is different.

Presently CyrTUG, the Cyrillic T_EX Users Group has a working group coordinated by Olga Lapko trying to come up with a unique encoding, but, as stated in the `babel` article, we will probably have to wait until the Ω system is ready to try and solve the problems discussed in Section “Encoding and font problems”.

Finally, as a technical aside, the picture on the next page, showing an example of SlavT_EX output, had its text typeset with the SlavT_EX system, while the first letter of the text and the surrounding frame are bitmaps.

References

- [1] Olga G. Lapko. MAKEFONT as part of the CyrTUG-EMT_EX package. *Proceedings of the 8th European T_EX Conference*, pages 110–118, Gdańsk, Poland, 1994.

◇ Michel Goossens
CERN, Geneva, Switzerland
Email: goossens@cern.ch

¹ More information on KOI8 can be found at the URL http://www.nar.com/tag/koi8_explained.html.

