Russian Encoding Plurality Problem and a New Cyrillic Font Set

L.N. Znamenskaya and S.V. Znamenskii
Krasnoyarsk State University, Svobodnyi prospekt 79, 660041 Krasnoyarsk, Russia
znamensk@ipsun.ras.ru

Abstract

To run TEX with cyrillic in network is a problem. Various widespread Cyrillic coding tables under DOS, UNIX and other OS are incompartible. The ASCII Russian text imported from a different system usually become completely unreadable. The new set of fonts, TDS and some other tools give a solution of the problem for the east-European Cyrillic typsertting users.

TEX has become the one of the best known means of communication between scientific people. To solve the problem of plural incompatible Russian TEX systems, the Russian Foundation for Basic Research (RFBR) proposed the idea of creation of a standard non-commercial Russian TEX distribution. Therefore, half a year ago the new "Russian TEX" project was begim under RFBR support. An important feature of the project is to determine the best system which is able to work in a LAN, with various client platforms and operating systems.

The new TDS (TEX Directory Structure) standard gives us the perfect base for a such system. The problem we find here is specific for the Cyrillic-based languages. It is the Russian encoding plurality problem. For example there exists several widely-used Russian coding tables under UNIX. Even Microsoft® uses completely different coding tables for Russian text under DOS and Windows® on the same PC. At the same time, in different directories on CTAN, we can find METAFONT sources for Cyrillic fonts with the same name `cmrz10` but with different Russian letter"A" character codes.

Fonts

The first thing we have had to do was to select an available Cyrillic extended standard TEX font set and fix new names in order to reflect a coding table in the name of font. As soon as we found the CyrTUG LH fonts not to be available for non-commercial RFBR distribution for free, we asked N. Glonty and A. Samarin for a permission to use their fonts, as they are the first and the most widely used TEX fonts in Russia. After a period of a month and a half, we received the very kind and grateful permission to use or modify the fonts or their sources for RFBR distribution and we appreciate very much such a generous solution. Unfortunately, we could not wait so long and and at this point in time, the development of the new Russian extension of a CM TEX font family was at the kerning stage. It so happened that we obtained the extra Russian extension of CM font family.

We tried to realise the following aims in this new font set:

- to keep the original CM font sources unchangeable to input by extension sources in order to provide appropriate Latin text when typesetting using the new fonts;
- to make text and letters more habitual for the Russian eye, keeping the traditional CM fonts peculiarity;
- to make letter darkness in text more uniform;
- to make all CM source based fonts, including `concrete` available for Russian typesetting;
- to avoid possible low-resolution font-creation errors causing problems while using automatic font generation; and
- to lay the foundation for future support of all Cyrillic-based alphabets of the Russian people.

We used CM macros, fragments of CM codes and a bit of `cmcyr` code. The acroLH font family has been used just for comparison in the first stage.

When the new fonts were almost ready it was decided to compare their typesetting quality with the one of the best sources of widely distributed fonts — the Samarin and Glonty Cyrillic fonts. A large mathematical paper has been printed at 10 and 12 points on a 600dpi HP LaserJet4 printer, the same text in two copies printed with different font sets. There was a blank page in each copy for experts to write their opinion. The RFBR experts (physicists and mathematicians) compared the two, and determined that the both Russian font families are of the same good quality.

What should we do with the new fonts names? The first idea was to use the fontname scheme. In this way, we made the name of extended 8-bit font much too different from the name of corresponding standard 7-bit CM font. As a result users would have a problems while adopting new styles and using the TeX primitive font selection commands. To reduce such problems we decided to create a font name from RF (Russian Font + Russian Foundation); to use the third char (digit) in the name to point to the coding table, and to end by using the same char sequence as that used by the corresponding CM font. One can see the examples on tables.

The empty boxes in font tables will be filled by other Cyrillic letters in next version of fonts. It is impossible to support all Cyrillic-based languages by the same 8-bit coding table — the number of different letters is more than 256. The project is working on a coding table which would allow typesetting on more than sixty Cyrillic-based languages with the use of accents or virtual fonts or `\charsubdef`. The list of languages to be supported in such a way contains all of the Cyrillic-based languages of Russia.

Russian encoding plurality problem

We need to support the typical situation of an entire TeX file system residing on a server, with clients working under different operation systems using various Russian encodings. The main problem is to select the appropriate procedure for inputting TeX files with *any* encoding.

Our way to solve this problem is to create an executable which would recognize the Cyrillic coding of a file in the correct way, and then recode it automatically to conform to the local coding.

Why not? Anybody who reads Russian can easily convert the text in the right coding from the same text in the wrong coding. But as soon as we try to look more carefully at the problem, we see the multiple problems.

The coding tables one-to-one correspondence as a part of problem If a binary file is occasionally to be recoded as Cyrillic, it is useful to have the capability of recovering an accidently-converted file. The networking forse problem to be more difficult: multiple convertions must preserve the original information. We cannot see a way to solve this problem without the additional difficulty of creating a proper conversion algorithm. It is natural to preserve the ASCII first 128 positions of code table. In the last 128 positions, we have to put one-to-one correspondence between each set of coding tables in a consistent way.

Unfortunately, this is not possible. The set of symbols in this part of the coding tables differs very much from one table to other. Therefore we have to permit the Rchar to change meaning during conversions. We try to considerably decrease the set of possible meaning changes. The desirable solution is to split the set of all possible char meanings of the 128 equivalence classes such that any conversion can change the symbol meaning only inside its equivalence class. This is also impossible. Some of the meanings will necessarily be found in different classes and the best thing we can do is to use the less valuable meanings for such a mess. You can see the summary of a various available information on Cyrillic coding tables [1]–[8] and our proposals on the one-to-one table correspondence in a huge table bellow. In this table the numbers 0, 1, 2, 3, 4, 5 respectively denote ISO8859-5, CP1251, PC866, -8, MacOS, and PC855.

The problem of other Cyrillic languages There are more then 60 Cyrillic-based languages and some of them still have not settled coding tables. Most of the files contains a lot of the non-text commands. There is a lot of software which puts a non-ASCII chars into file and the program has to distinguish, as far as is possible, the right Cyrillic words from the combinations of such symbols.

We therefore cannot use only the char set information of the file to discover the coding table of document. Another problem we see is that some coding tables use the same char set. As we need to get a right solution for a short file, it is also inadequate just to count the number of each letters appearing in text. A more precise instrument would be to count the number of each combinations of two letters appearing in the document.

This effective approach require more them 128 kilobytes of memory for an intermediate data storage. The natural algorithm to perform a proper statistical analysis of this data includes multiple computing of logarithms and is not fast enough – especially on a PC. How to find a way to get the acceptable result in a simple and fast way?

The next idea was to select two sets of possible strings of length 2: the set, A, of frequently-appearing Cyrillic text bicharacter strings and a set, U, of commonly unused Cyrillic text bicharacter strings. The executable counts the numbers N_A and N_U of strings from A and U, respectively, appearing in the file. The number $C = \frac{N_A - N_U}{N_A + N_U}$ will show if this file looks as Cyrillic text or not. Such a number can

where	THE MEANING
23	BOX DRAWINGS DOWN SINGLE AND RIGHT DOUBLE
0145	CYRILLIC CAPITAL LETTER DJE
23	RIGHT HALF BLOCK
0145	CYRILLIC CAPITAL LETTER GJE
23	BOX DRAWINGS DOWN SINGLE AND LEFT DOUBLE
0145	CYRILLIC CAPITAL LETTER DZE
23	LEFT HALF BLOCK
0145	CYRILLIC CAPITAL LETTER BYELORUSSIAN-UKRAINIAN I
3	TOP HALF INTEGRAL
01245	CYRILLIC CAPITAL LETTER YI
23	BOX DRAWINGS UP SINGLE AND RIGHT DOUBLE
0145	CYRILLIC CAPITAL LETTER JE
23	BOX DRAWINGS UP DOUBLE AND RIGHT SINGLE
0145	CYRILLIC CAPITAL LETTER LJE
23	BOX DRAWINGS UP SINGLE AND LEFT DOUBLE
0145	CYRILLIC CAPITAL LETTER NJE
23	BOX DRAWINGS UP DOUBLE AND LEFT SINGLE
0145	CYRILLIC CAPITAL LETTER TSHE
23	BULLET OPERATOR
0145	CYRILLIC CAPITAL LETTER KJE
23	BOX DRAWINGS VERTICAL SINGLE AND RIGHT DOUBLE
0145	CYRILLIC CAPITAL LETTER DZHE
23	BOX DRAWINGS VERTICAL DOUBLE AND RIGHT SINGLE
0145	CYRILLIC SMALL LETTER DJE
23	BOX DRAWINGS VERTICAL SINGLE AND LEFT DOUBLE
0145	CYRILLIC SMALL LETTER GJE
23	BOX DRAWINGS VERTICAL DOUBLE AND LEFT SINGLE
0145	CYRILLIC SMALL LETTER DZE
23	BOX DRAWINGS DOWN SINGLE AND HORIZONTAL DOUBLE
0145	CYRILLIC SMALL LETTER BYELORUSSIAN-UKRAINIAN I
3	BOTTOM HALF INTEGRAL
01245	CYRILLIC SMALL LETTER YI
23	BOX DRAWINGS DOWN DOUBLE AND HORIZONTAL SINGLE
0145	CYRILLIC SMALL LETTER JE
23	BOX DRAWINGS UP SINGLE AND HORIZONTAL DOUBLE
0145	CYRILLIC SMALL LETTER LJE
23	BOX DRAWINGS UP DOUBLE AND HORIZONTAL SINGLE
0145	CYRILLIC SMALL LETTER NJE
23	BOX DRAWINGS VERTICAL SINGLE AND HORIZONTAL DOUBLE
0145	CYRILLIC SMALL LETTER TSHE
23	BOX DRAWINGS VERTICAL DOUBLE AND HORIZONTAL SINGLE
0145	CYRILLIC SMALL LETTER KJE
23	FULL BLOCK *)
0145	CYRILLIC SMALL LETTER DZHE
235	BOX DRAWINGS LIGHT VERTICAL AND RIGHT
14	CYRILLIC CAPITAL LETTER GHE WITH UPTURN
235	BOX DRAWINGS LIGHT VERTICAL AND LEFT
14	CYRILLIC SMALL LETTER GHE WITH UPTURN

Table 1: the non-russian letters
*) for this coding table

WHERE	THE MEANING
235	BOX DRAWINGS LIGHT UP AND RIGHT
14	LEFT SINGLE QUOTATION MARK
235	BOX DRAWINGS LIGHT UP AND LEFT
14	RIGHT SINGLE QUOTATION MARK
235	BOX DRAWINGS DOUBLE UP AND LEFT
14	LEFT DOUBLE QUOTATION MARK
235	BOX DRAWINGS DOUBLE UP AND RIGHT
14	RIGHT DOUBLE QUOTATION MARK
235	BOX DRAWINGS DOUBLE DOWN AND LEFT
14	DOUBLE LOW-9 QUOTATION MARK
4	POUND SIGN
235	BOX DRAWINGS LIGHT DOWN AND RIGHT
1	SINGLE LOW-9 QUOTATION MARK

Table 2: the symbols look more-or-less like left/right coma quotation

WHERE	THE MEANING
3	GREATER-THAN OR EQUAL TO *)
01245	CYRILLIC CAPITAL LETTER UKRAINIAN IE
3	DIVISION SIGN *)
01245	CYRILLIC CAPITAL LETTER SHORT U
3	LESS-THAN OR EQUAL TO *)
01245	CYRILLIC SMALL LETTER UKRAINIAN IE
3	ALMOST EQUAL TO *)
01245	CYRILLIC SMALL LETTER SHORT U

Table 3: pc855/pc866 splittings
*) for this coding table

WHERE	THE MEANING
23	BOX DRAWINGS DOWN DOUBLE AND LEFT SINGLE
145	LEFT-POINTING DOUBLE ANGLE QUOTATION MARK
23	BOX DRAWINGS DOWN DOUBLE AND RIGHT SINGLE
145	RIGHT-POINTING DOUBLE ANGLE QUOTATION MARK
4	LESS-THAN OR EQUAL TO *)
235	BOX DRAWINGS DOUBLE VERTICAL AND LEFT
1	SINGLE LEFT-POINTING ANGLE QUOTATION MARK
4	GREATER-THAN OR EQUAL TO *)
235	BOX DRAWINGS DOUBLE VERTICAL AND RIGHT
1	SINGLE RIGHT-POINTING ANGLE QUOTATION MARK

Table 4: the symbols look more-or-less like left/right angle quotation
*) for this coding table

where	the meaning
3	superscript two
01245	numero sign
23	middle dot *)
0145	section sign
25	lower half block *)
134	copyright sign
235	box drawings light vertical
14	not sign
235	box drawings light vertical and horizontal
14	registered sign
235	box drawings light down and left
14	plus-minus sign
235	box drawings double horizontal
14	micro sign
235	box drawings double vertical
14	pilcrow sign
235	box drawings light down and horizontal
14	en dash
235	box drawings light up and horizontal
14	em dash
235	box drawings light horizontal
14	dagger
235	box drawings double down and right
14	bullet
235	light shade
14	horizontal ellipsis
235	box drawings double down and horizontal
14	trade mark sign
4	not equal to
235	box drawings double up and horizontal
1	double dagger
4	infinity
235	box drawings double vertical and horizontal
1	not used
4	increment
235	upper half block
1	per mille sign
012345	no-break space
4	division sign *)
235	medium shade
1	broken bar
4	latin small letter f with hook
235	dark shade
1	middle dot *)
4	almost equal to *)
235	black square
1	not used
5	full block *)
1234	degree sign
234	square root
015	soft hyphen
3	lower half block *)
1245	currency sign

Table 5: other symbols
*) for some coding tables

be computed for each known coding table and the largest value must point to the right coding table. It seems to be fast, easy and effective because the most frequently used conjunctions of two characters (less then 5% of all conjunctions) gives more then 50% of bicharacter substrings in Russian text and approximately half of all possible conjunctions which are practically never used in Russian. The "only" problem remaining is to select the sets A and U properly.

How we selected A and U A great help for us was the unique Gilyarovskii and Grivnin book [9] with the text samples on most of the languages. We had to turn the samples into computer files in order to count biletter appearance numbers. A new problem then arose: what should we do with non-Russian letters?

There are no fixed coding tables for most of the languages. We also do not know about any other attempts to use a Russian keyboard and special TEX commands for typesetting of most of the Cyrillic languages of Russia, Mongolia and Alaska. For each of the languages which use non-Russian letters, we have made two files: the first file has char representation of non-Russian letters mostly according to the tables above, and the second file has more-or-less better readable Russian letter sequences following the slash char (such as /ЪК for "K as in beak" or /КЦ for "K as in desk" or /ЛЬ for Љ or /Ц for Џ) and maximal usage of the standard TEX accent control sequences. For the Russian language, we used three different subject topics and a dictionary with 51924 words. Each of the other languages was represented by a single file. We obtained 109 files for 64 languages.

We cannot be certain other people will use the same codes or sequences for non-Russian letters. Therefore, while counting the biletter strings for each file we assign all letters with unknown codes to a group, identify all ASCII non-letters and assign them to another group and assign all Latin letters unusable by Cyrillic text to a separate group. After, counting we selected biletter strings which did not appeared in files. They composed the set U with 695 elements.

The selection of set A was more difficult. After several attempts to select it we got the following algorithm. For each couple of letters and each file, the logarithm of 'relative frequence' was computed. To avoid infinity we had zero frequences changed to a small non-zero value, as if this biletter string appears once in a file twice as long. Then we found the sums over all the files and used them for selection. The most frequent 314 couples consist of only Russian letters and almost each word contains at least one of such biletter strings. We had to avoid the effects of possible usage of other TEX names for non-Russian letters, or other coding tables which may correlate only to the Russian part of our coding table. Therefore we used only 306 of these couples without the biletter strings which our special notations for non-russian letters could produce.

In this way, the Cyrillic coding recognition algorithm was finished.

Availability

The METAFONT sources of RF font family and sources of cyrillic coding recognition algorithm will be available from RFBR TEX server via anonymous ftp: `ftp.tex.math.ru`.

Acknowledgements

This work was inspired and supported by Russian Foundation for Basic Research, grant 96-07-89406.

References

[1] A. Chernov. Registration of a Cyrillic Character Set. RFC 1489, RELCOM Development Team, July 1993.

[2] J. Reynolds, J. Postel. Assigned Numbers. RFC 1700, USC/Information Sciences Institute, October 1994.

[3] T.Greenwood, J. H. Jenkins. ISO 8859-5 (1988) to Unicode. Unicode Inc. January 1995.

[4] M. Siugnard, L. Hoerth. cp1251_WinCyrillic to Unicode table. Unicode Inc. March 1995.

[5] M. Siugnard, L. Hoerth. cp10007_MacCyrillic to Unicode table. Unicode Inc. March 1995.

[6] M. Siugnard, L. Hoerth. cp855_DOSCyrillic to Unicode table. Unicode Inc. March 1995.

[7] M. Siugnard, L. Hoerth. cp866_DOSCyrillicRussian to Unicode table. Unicode Inc. March 1995.

[8] P. Edberg. MacOS_Ukrainian [to Unicode]. Unicode Inc. April 1995.

[9] Р.С. Гиляровский, В.С. Гривнин. Определитель языков мира по письменности. Изд-е третье, исправленное и дополненное. М.: Наука, 1964.