# Cyrillic TEX files: interplatform portability

Peter A. Ovchenkov
Russia, Moscow 125047, Miusskaya sq. 4, M. V. Keldysh Institute of Applied Mathematics
ptr@ASoft.MSK.SU

### Abstract

We consider the problems with regard to Cyrillic encoding while working with different operatining systems. The proposed solutions can simplify the administration and support of TEX on different platforms. The related problem of encoding Cyrillic TEX fonts is also discussed.

## Preliminary notes

A number of new Cyrillic encodings have appeared recently. Maybe that's all? But no — the process needs further consideration. Why, indeed, ... is Bill Gates better than me? I would like to suggest trying Cyrillic encoding for TEX fonts. If you don't like it, there are almost as many TEX fonts as Cyrillic encodings. If you multiply these two numbers, you can see fantastic potential for encodings suggestions.

I would also like to talk about the native Russian typesetting, not translitaration. The examples I will use demonstrate only Russian letters; I do not have enough experience to talk about letters specific to other Cyrillic alphabets.

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | a | b | c | d | e | f |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x | | | | | | | | | | | | | | | | |
| 1x | | | | | | | | | | | | | | | | |
| 2x | ␣ | ! | " | # | $ | % | & | ’ | ( | ) | * | + | , | - | . | / |
| 3x | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | : | ; | < | = | > | ? |
| 4x | @ | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O |
| 5x | P | Q | R | S | T | U | V | W | X | Y | Z | [ | \ | ] | ˆ | _ |
| 6x | ‘ | a | b | c | d | e | f | g | h | i | j | k | l | m | n | o |
| 7x | p | q | r | s | t | u | v | w | x | y | z | { | \| | } | ˜ | |
| 8x | А | Б | В | Г | Д | И | Ж | З | И | Й | К | Л | М | Н | О | П |
| 9x | Р | С | Т | У | Ф | Х | Ц | Ч | Ш | Щ | Ъ | Ы | Ь | Э | Ю | Я |
| ax | а | б | в | г | д | е | ж | з | и | й | к | л | м | н | о | п |
| bx | | | | | | | | | | | | | | | | |
| cx | | | | | | | | | | | | | | | | |
| dx | | | | | | | | | | | | | | | | |
| ex | р | с | т | у | ф | х | ц | ч | ш | щ | ъ | ы | ь | э | ю | я |
| fx | Ё | ё | | | | | | | | | | | | | | |

**Figure 1**: Cyrillic encoding “alternative” (code page 866).

## Problem

First of all, for historical reasons, many Cyrillic encodings were developed for different types of computers (and different operating systems). To simplify, I will speak only about computers with 8 bits/byte and with Latin characters encoded as US-ASCII. For such machines, at least four cyrillic encodings exist. For computers running DOS, the most widely used encoding is the so-called alternative encoding (fig. 1). Its popularity here is because the pseudographic symbols have the same codes as those for extended ASCII. For MS Windows, the original Microsoft encoding “honored Bill Gates” (fig. 2). On UNIX machines two encodings coexist — KOI-8 (fig. 3) and ISO 8859-5 (fig. 4). ISO 8859-5 is the official standard (GOST). Macintoshes use the ISO 8859-5 standard. That's enough without Cyrillic in EBCDIC-like encodings.

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | a | b | c | d | e | f |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x | | | | | | | | | | | | | | | | |
| 1x | | | | | | | | | | | | | | | | |
| 2x | ␣ | ! | " | # | $ | % | & | ’ | ( | ) | * | + | , | - | . | / |
| 3x | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | : | ; | < | = | > | ? |
| 4x | @ | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O |
| 5x | P | Q | R | S | T | U | V | W | X | Y | Z | [ | \ | ] | ˆ | _ |
| 6x | ‘ | a | b | c | d | e | f | g | h | i | j | k | l | m | n | o |
| 7x | p | q | r | s | t | u | v | w | x | y | z | { | \| | } | ˜ | |
| 8x | | | | | | | | | | | | | | | | |
| 9x | | | | | | | | | | | | | | | | |
| ax | | | | | | | | | Ё | | | | | | | |
| bx | | | | | | | | | ё | № | | | | | | |
| cx | А | Б | В | Г | Д | Е | Ж | З | И | Й | К | Л | М | Н | О | П |
| dx | Р | С | Т | У | Ф | Х | Ц | Ч | Ш | Щ | Ъ | Ы | Ь | Э | Ю | Я |
| ex | а | б | в | г | д | е | ж | з | и | й | к | л | м | н | о | п |
| fx | р | с | т | у | ф | х | ц | ч | ш | щ | ъ | ы | ь | э | ю | я |

**Figure 2**: Cyrillic encoding in MS Windows (code page 1251).

Let us consider the following problem: you need to process TEX files on computers with different Cyrillic encodings. Don't worry about your `.tex` files: (i) there are enough transcoder programs, and (ii) there are enough program tools (like text editors, etc.) that are outside TEX's control. But

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | a | b | c | d | e | f |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x | | | | | | | | | | | | | | | | |
| 1x | | | | | | | | | | | | | | | | |
| 2x | ␣ | ! | " | # | $ | % | & | ’ | ( | ) | * | + | , | - | . | / |
| 3x | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | : | ; | < | = | > | ? |
| 4x | @ | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O |
| 5x | P | Q | R | S | T | U | V | W | X | Y | Z | [ | \ | ] | ^ | _ |
| 6x | ‘ | a | b | c | d | e | f | g | h | i | j | k | l | m | n | o |
| 7x | p | q | r | s | t | u | v | w | x | y | z | { | \| | } | ~ | |
| 8x | | | | | | | | | | | | | | | | |
| 9x | | | | | | | | | | | | | | | | |
| ax | | | | ё | | | | | | | | | | | | |
| bx | | | | Ё | | | | | | | | | | | | |
| cx | ю | а | б | ц | д | е | ф | г | х | и | й | к | л | м | н | о |
| dx | п | я | р | с | т | у | ж | в | ь | ы | з | ш | э | щ | ч | ъ |
| ex | Ю | Б | В | Ц | Д | Е | Ф | Г | Х | И | Й | К | Л | М | Н | О |
| fx | П | Я | Р | С | Т | У | Ж | В | Ь | Ы | Ч | Ш | Э | Щ | Ч | Ъ |

**Figure 3**: Cyrillic encoding KOI8.

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | a | b | c | d | e | f |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x | | | | | | | | | | | | | | | | |
| 1x | | | | | | | | | | | | | | | | |
| 2x | ␣ | ! | " | # | $ | % | & | ’ | ( | ) | * | + | , | - | . | / |
| 3x | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | : | ; | < | = | > | ? |
| 4x | @ | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O |
| 5x | P | Q | R | S | T | U | V | W | X | Y | Z | [ | \ | ] | ^ | _ |
| 6x | ‘ | a | b | c | d | e | f | g | h | i | j | k | l | m | n | o |
| 7x | p | q | r | s | t | u | v | w | x | y | z | { | \| | } | ~ | |
| 8x | | | | | | | | | | | | | | | | |
| 9x | | | | | | | | | | | | | | | | |
| ax | | Ё | | | | | | | | | | | | | | |
| bx | А | Б | В | Г | Д | И | Ж | З | И | Й | К | Л | М | Н | О | П |
| cx | Р | С | Т | У | Ф | Х | Ц | Ч | Ш | Щ | Ъ | Ы | Ь | Э | Ю | Я |
| dx | а | б | в | г | д | е | ж | з | и | й | к | л | м | н | о | п |
| ex | р | с | т | у | ф | х | ц | ч | ш | щ | ъ | ы | ь | э | ю | я |
| fx | № | ё | | | | | | | | | | | | | | |

**Figure 4**: Cyrillic encoding ISO 8859-5.

what about `.dvi` files and style files? Oh, this is the chief problem for the TEX administrator in a heterogeneous network. He (or she) must:

1. define as letters (catcode 11) Cyrillic letters for all encodings in usage (during format generation)
2. transcode styles with Cyrillic styles (this would preclude sharing LATEX styles via an electronic network)
3. replace fonts with virtual fonts

The first two operations are simple, but if you forget to convert a style to a Cyrillic style, for example, you discover unusual output: and in some cases, the mistake may be very difficult to locate.

Remapping fonts into virtual fonts is also not without problems: for example, what do you do with the kerning between punctuation (codes less than 127) and Cyrillic letters (codes greater than 128)? The use of 8-bit full fonts leads to incompatibility of `.dvi` files prepared on different platforms and dramatically increases the number of fonts.

## Input Flow Transformation

Let us consider a hypothetical 8-bit font: the first half is Latin symbols (symbols with codes $00_h$–$79_h$), punctuation, and digits, etc., and the second half contains Cyrillic letters. Let define the font encoding as $E_f$. For TEX (the binary executable program) we can define a transformation $\kappa$, such that $E_e \xrightarrow{\kappa} E_i$, where $E_e$ is a character encoding that TEX "sees" as input flow (i.e., "native" encoding of a (soft)hardware system), and $E_i$ — TEX internal encoding.

If we define $E_i \equiv E_f$, we can use the same fonts during editing and previewing on one machine type, and printing on another one. (This is obvious for systems with Latin typesettings, but not for Cyrillic!) We will had that a `.dvi` file (created on a PC) can be converted by dvips on SPARCstation without problems.

In the input flow you can refer directly to a symbol in "internal" encoding (in $E_i$ representation). This fact allows one to create style files that can be used on different platforms simultaneously.

**Input Flow Transformation in emTEX** In emTEX, Eberhard Mattes suggests a simple way to set up $\kappa$ transformation. This is the creation of TEX Code Page to then include this information in the precompiled format file.

The transformation is described in the `.mtc` file. Below is the beginning of the file for the alternative — experimental (compare figures 1 and 5) encoding:

```
%
% alt_abs.mtc
%
^^80 ^^c0
^^81 ^^c1
^^82 ^^c2
^^83 ^^c3
^^84 ^^c4
^^85 ^^c5
^^86 ^^c6
^^87 ^^c7
^^88 ^^c8
^^89 ^^c9
^^8a ^^ca
^^8b ^^cb
^^8c ^^cc
^^8d ^^cd
^^8e ^^ce
^^8f ^^cf
```

Here, the first column represents the character code that TEX see as input flow, and second is the one inside TEX. In the second column you can write TEX commands instead of character codes.

Next the `.mtc` file is compiled into the `.tcp`file:

```
C:\EMTEX\DATA> maketcp -c -8 alt_abs.mtc
```

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | a | b | c | d | e | f |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x | ` | ´ | ˆ | ˜ | ¨ | ˝ | ˚ | ˇ | ˘ | ¯ | ˙ | ¸ | ˛ | ‚ | ‹ | › |
| 1x | “ | ” | „ | « | » | – | — | | ₀ | ı | ȷ | ﬀ | ﬁ | ﬂ | ﬃ | ﬄ |
| 2x | ␣ | ! | " | # | $ | % | & | ’ | ( | ) | * | + | , | - | . | / |
| 3x | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | : | ; | < | = | > | ? |
| 4x | @ | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O |
| 5x | P | Q | R | S | T | U | V | W | X | Y | Z | [ | \ | ] | ˆ | _ |
| 6x | ‘ | a | b | c | d | e | f | g | h | i | j | k | l | m | n | o |
| 7x | p | q | r | s | t | u | v | w | x | y | z | { | \| | } | ˜ | - |
| 8x | | | | | | Ё | | | | | | | | | | |
| 9x | | | | | | | | | | | | | | | | § |
| ax | | | | | | ё | | | | | | | | | | |
| bx | | | | | | | | | | | № | | | ¡ | ¿ | £ |
| cx | А | Б | В | Г | Д | Е | Ж | З | И | Й | К | Л | М | Н | О | П |
| dx | Р | С | Т | У | Ф | Х | Ц | Ч | Ш | Щ | Ъ | Ы | Ь | Э | Ю | Я |
| ex | а | б | в | г | д | е | ж | з | и | й | к | л | м | н | о | п |
| fx | р | с | т | у | ф | х | ц | ч | ш | щ | ъ | ы | ь | э | ю | я |

**Figure 5**: Experimental font encoding ($E_i \equiv E_f$).

The code table is turned on while the format file is being created:

```
C:\EMTEX\LATEX\BASE> latex -i -8 \
   -c alt_abs -mt15000 -mp65500 latex.ltx
```

There are some other options that you can find in the emTEX documentation.

**Input flow transformation in UNIX-like systems** On UNIX-like systems, TEX is traditionally compiled from WEB sources. In many cases this is done via CWEB. Here it seems there is no simple way, as there is for emTEX. One is forced to program; but for UNIX you always can find a C-compiler (at least).

While TEX is translated from WEB to C, there are applied patches from the file ctex.ch. You can create your own variant of this file with minor additions.

First we remap the input flow (the example below illustrates the remapping from ISO 8859-5 encoding into the experimental one – see figures 4, 5):

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% [2.23] Allow any character as input.
%        (Remapping by ptr)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
@x
for i:=0 to @'37 do xchr[i]:=' ';
for i:=@'177 to @'377 do xchr[i]:=' ';
@y
for i:=0 to @'37 do xchr[i]:=chr(i);
for i:=@'177 to @'237 do xchr[i]:=chr(i);
for i:=@'240 to @'357 do
                      xchr[i+16]:=chr(i);
for i:=@'360 to @'377 do
                      xchr[i-64]:=chr(i);
xchr[@'205]:=chr(@'241);{\Yo}
xchr[@'245]:=chr(@'361);{\yo}
@z
```

Substitute printable symbols for the non-printable ones, (i.e., TEX substitutes on the terminal display, and xwrites in the .log file:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% [?.??] Modify characters cannot be
%        printed -- Ptr.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
@x
@<Character |k| cannot be printed@>=
  (k<" ")or(k>"~")
@y
@<Character |k| cannot be printed@>=
  (k<" ")or((k>"~")and(k<@'300)
          and(k<>@'270)and(k<>@'271))
@z
```

After that you can build TEX in the normal way.

### Fonts Encoding

But there is one small thing: now we have as many font encodings as machine ones. Now's the time to find a uniform Cyrillic font encoding.

The problem connected with letters of national alphabets in Europe was solved in part with unification of TEX fonts. The arguments, pro and contra, connected with the Cork encoding scheme, you can find in various *TUGboat*s, or in the *LaTeX Companion*.

There is no more room for Cyrillic in the Cork encoding. That's evident. But the experience of dc-fonts is good (in my opinion). So I think that a good solution is encoding similar to that in figure 5. The first half is equivalent to the first half of the Cork encoding, and Cyrillic is placed in the second half. Why is there no ISO- or alternative-like encoding here? This is due to the LaTeX $2_\varepsilon$ requirement for the difference between codes for upper- and lowercase of the same letter to be 32, and a sequence of letters beginning with code 128 or 192. There is no evident preference for any machine encoding of Cyrillic, but with an encoding as seen in figure 5 we can avoid some of the problems with LaTeX $2_\varepsilon$.

### Cyrillic in Style Files

TEX allows direct references on internal TEX encoding (^^hh, hh is a pair of hex digits). This fact can be used for writing files for the precompiled format or for styles: such files depend only upon font encoding, but not upon the Cyrillic encoding on computer.

For precompiled format files we can write definitions for Russian letters and then use this file on every computer:

```
\def\letter#1#2{%
    \catcode`#1=11\catcode`#2=11%
    \uccode`#1=`#1\lccode`#1=`#2%
    \uccode`#2=`#1\lccode`#2=`#2%
}
\language=\l@russian
\letter{^^c0}{^^e0}%
\letter{^^c1}{^^e1}%
\letter{^^c2}{^^e2}%
\letter{^^c3}{^^e3}%
```

Or, we can do the same with styles:

```
\addto\captionsrussian{%
  \def\prefacename{%
^^cf^^f0^^e5^^e4^^e0^^f1^^eb^^ee^^e2%
^^e8^^e5}%
  \def\refname{%
^^cb^^e8^^f2^^e5^^f0^^e0^^f2^^f3^^f0^^e0}
  \def\abstractname{%
^^c0^^ed^^ed^^ee^^f2^^e0^^f6^^e8^^ff}%
  \def\bibname{%
^^c1^^e8^^e1^^e8^^eb^^e8^^ee^^e3^^f0%
^^e0^^f4^^e8^^ff}%
  \def\chaptername{^^c3^^eb^^e0^^e2^^e0}%
  \def\appendixname{%
^^cf^^f0^^e8^^eb^^ee^^e6^^e5^^ed^^e8^^e5}
  \def\contentsname{%
^^d1^^ee^^e4^^e5^^f0^^e6^^e0^^ed^^e8^^e5}
  \def\listfigurename{%
^^d1^^ef^^e8^^f1^^ee^^ea %
^^f0^^e8^^f1^^f3^^ed^^ea^^ee^^e2}%
  \def\listtablename{%
^^d1^^ef^^e8^^f1^^ee^^ea %
^^f2^^e0^^e1^^eb^^e8^^f6}%
  \def\indexname{^^c8^^ed^^e4^^e5^^ea^^f1}
  \def\figurename{%
^^d0^^e8^^f1^^f3^^ed^^ee^^ea}%
  \def\tablename{%
^^d2^^e0^^e1^^eb^^e8^^f6^^e0}%
  \def\partname{^^d7^^e0^^f1^^f2^^fc}%
  \def\enclname{^^e2^^ea^^eb.}%
  \def\ccname{^^e8^^e7}%
  \def\headtoname{^^e2}%
  \def\pagename{%
^^f1^^f2^^f0^^e0^^ed^^e8^^f6^^e0}%
  \def\seename{^^f1^^ec.}%
  \def\alsoname{^^f1^^ec.~^^f2^^e6.}%
}
```

Of course, this is not very readable, but this is done by a style developer, and only once for every machine-specific encoding of Cyrillic fonts.

One note: this mechanism cannot be applied to hyphenation patterns. TEX doesn't expect $E_i$-symbols representation (`^^hh`) inside command `\patterns`.