

The Linux Cyrillic HOWTO (russian)

Евгений Балдин

<E.M.Baldin@inp.nsk.su>

Версия 1.4.2 от 29 сентября 2002 г.

Аннотация

Здесь описывается, как настроить кириллицу для редактирования, просмотра и печати документов в Unix подобных системах^[1].

Глава 1. Введение

Содержание

В этом документе описываются приёмы, которые нужны для редактирования, просмотра и печати документов с использованием кириллических символов под Linux. И хотя здесь предполагается, что вы используете Linux как операционную систему, большая часть информации одинаково применима и к другим разновидностям Unix.

Предполагается, что любая операционная система UNIX настраивается и поддерживается опытным человеком. Одного знакомства с книгой из серии "что-то там для чайников" недостаточно. Кириллизация операционной системы модифицирует её, следовательно, требует определённых знаний о том, что вы делаете. Несмотря на то, что я пробовал упростить изложение настолько, насколько это возможно, наличие некоторого опыта работы с настраиваемым программным обеспечением является преимуществом на пути к "тотальной кириллизации". В этом тексте нет описания что такое X Window, как создаются документы в системах TeX и LaTeX или как установить и настроить принтер в Linux. Это описано в других документах. UNIX — это многопользовательская система, и поэтому условия, при которых вы проводите кириллизацию, могут отличаться. Вы можете быть системным администратором (или владельцем системы), пытающимся кириллизировать всю систему. С другой стороны, вы можете быть обычным пользователем, не имеющим привилегий системного администратора (root или superuser), и вы хотите кириллизировать систему только для себя. Большинство описываемых мной программ обладают достаточной гибкостью, позволяющей настроить их и с точки зрения всей системы, и с точки зрения обычного пользователя. Я попытаюсь отметить оба случая.

Перед тем, как настроить различные части системы, следует настроить пару простых вещей. Большинство программных компонент, описываемых ниже, предполагают, что кириллические шрифты доступны и пользователь может вводить кириллические символы. Чтобы это действительно стало правдой, следует настроить окружение для того, чтобы обеспечить и шрифты, и возможность ввода кириллицы.

Linux поддерживает два интерфейса для отображения информации — два различных режима. Один из них — это текстовый режим (консоль), а другой — графический, предоставляемый средствами X Window. Оба этих режима требуют разной настройки, которая описывается в этом документе.

ВНИМАНИЕ: X Window, TeX и другие компоненты Linux — это сложные системы. Если вы сделаете что-либо неправильно, то вы не только потерпите неудачу с кириллизацией системы, но и можете частично, если не полностью, испортить систему. Это не должно вас пугать, это просто предупреждение, которое даёт вам понять всю серьёзность процесса настройки такого типа. Очень рекомендуется сделать копии файлов конфигурации. Также неплохо бы иметь под рукой гуру (специалиста по Unix) — просто так, на всякий пожарный.

Местоположение документа и обратная связь

Последнюю версию этого документа всегда можно найти [на моей домашней странице](#) в разделе Cyrillic HOWTO (russian).

Старую версию документа на английском языке, которую создал А.Беликов, можно найти на www.ldp.org или на tsx-11.mit.edu как часть *Linux Document Project* в разделе *unmantained*. Также его можно найти на различных FTP, имеющих отношение к Linux. Кроме того, этот документ может быть включён в дистрибутив Linux как его часть.

Дословный перевод устаревшей версии можно найти на [на моей домашней странице](#) в разделе Cyrillic HOWTO (old rus).

Если у вас имеются какие-либо предложения или исправления, имеющие отношение к этому документу, то, пожалуйста, незамедлительно свяжитесь со мной по адресу E.M.Baldin@inp.nsk.su. За предоставление любой

новой и полезной информации относительно поддержки кириллицы в различных Unix'ах буду *крайне* признателен. Не забудьте, ведь это поможет другим.

Исходным форматом документа является *XML*. В случае внесения большого количества исправлений лучше ориентироваться на исходники: [Cyrillic-HOWTO-russian.tar.bz2](#). Исходный документ разбит на несколько частей.

Благодарность and copyrights

Много людей помогли мне (и не только мне) ценной информацией и предложениями. И даже большее количество людей создало программное обеспечение для public community. Мне очень жаль, если я забыл кого-то упомянуть.

Итак, признательность нашим героям — это: св. Кирилл, св. Мефодий, Александр Беликов, Андрей Беляев, Дмитрий Болховитянов, Ф.А. Брокгауз, Евгений Бырганов, Виктор Вагнер, Александр Васильев Александр Валишев, Владимир Волович, Александр Воронов, Александр Воропаев, Игорь Власенко, Владимир Дмитриев, И.А. Ефрон, Андрей Ёлкин, Александр Жуков, Антон Зиновьев, Mikhail Zislis, Евгений Коробейников, Александр Лебедев, Алексей Махоткин, Александр Михаилян, Алексей Мичурин, Ольга Лапко (за шрифты lh), Леон Кантер, Алексей Киселёв, Михаил Колодин, Евгений Краус, Mykola Lyakhovych, Алексей Новодворский, Алексей Отт, Иван Паскаль, Фёдор Подгорный, Дмитрий Рубин, Андрей Соболев, Сергей Серебро, Дмитрий Сивашенко, Владимир Стусь, Борис Тоботрас, Дмитрий Фёдоров, Андрей Чернов, Андрей Юшков и другие — имя коим "отличные парни" (к сожалению, только парни) из [relcom.fido.ru.unix](#) и [relcom.fido.ru.linux](#) Usenet групп новостей (если кто-то хочет, чтобы я расписал поподробнее о том, за что я ему благодарен — дайте мне знать, и я это сделаю).

Cyrillic HOWTO (rus) — Copyright (C) 2000, 2001, 2002 Балдина Евгения Михайловича. Он может использоваться и распространяться под обычными Linux HOWTO условиями, описанными ниже.

В качестве основы при его создании был взят Cyrillic HOWTO — Copyright (C) 1995,1997 Александра Л. Беликова.

Далее идёт примечание к авторским правам Linux HOWTO:

Если не оговорено иначе, документы серии Linux HOWTO защищены авторскими правами их авторов. Документы серии Linux HOWTO могут воспроизводиться и распространяться полностью или частично, любым физическим или электронным способом, покуда это заявление авторского права сохраняется во всех копиях. Коммерческое распространение позволяет и поощряется, однако автора следует оповещать относительно любых подобных действий.

Все работы, использующие данный документ, включая любой документ серии Linux HOWTO, должны распространяться под этим же соглашением. То есть вы не можете накладывать дополнительные ограничения на распространение своего продукта, основанного на данном HOWTO. Исключения из этих правил могут предоставляться только при некоторых условиях; пожалуйста, войдите в контакт с координатором проекта Linux HOWTO по адресу, указанному ниже.

Другими словами, мы желаем поддержать распространение этой информации через максимально возможное количество каналов распространения. Однако, мы желаем сохранить авторские права на документы серии HOWTO, и хотелось бы, чтобы нас предупреждали о любых планах относительно распространения этого HOWTO.

Если у вас имеются вопросы, пожалуйста, войдите в контакт с Tim Bynum, координатором проекта Linux HOWTO, по адресу linux-howto@sunsite.unc.edu. Для того, чтобы получить номер телефона и дополнительную информацию для контакта, вы можете "напустить" на этот адрес finger.

Unix — торговая марка X/Open Ltd.; MS-DOS, Windows, Windows 95 и Windows NT — торговые марки Microsoft Corp.; X Window System — торговая марка X Consortium Inc. Другие торговые марки принадлежат соответствующим владельцам.

Ниже идёт оригинальный текст соглашения об авторских правах, который следует включать во все распространяемые копии этого документа и документов, созданных на его основе.

This document is Copyright (C) 2000, 2001, 2002 by Evgeni M. Baldine. It may be used and distributed under the usual Linux HOWTO terms described below.

The following is a Linux HOWTO copyright notice:

Unless otherwise stated, Linux HOWTO documents are copyrighted by their respective authors. Linux HOWTO documents may be reproduced and distributed in whole or in part, in any medium physical or electronic, as long as this copyright notice is retained on all copies. Commercial redistribution is allowed and encouraged; however, the author would like to be notified of any such distributions.

All translations, derivative works, or aggregate works incorporating any Linux HOWTO documents must be covered under this copyright notice. That is, you may not produce a derivative work from a HOWTO and impose additional restrictions on its distribution. Exceptions to these rules may be granted under certain conditions; please contact the Linux HOWTO coordinator at the address given below.

In short, we wish to promote dissemination of this information through as many channels as possible. However, we do wish to retain copyright on the HOWTO documents, and would like to be notified of any plans to redistribute the HOWTOS.

If you have questions, please contact Tim Bynum, the Linux HOWTO coordinator, at linux-howto@sunsite.unc.edu. You may finger this address for phone number and additional contact information. Unix is a technology trademark of the X/Open Ltd.; MS-DOS, Windows, Windows 95, and Windows NT are trademarks of the Microsoft Corp.; The X Window System is a trademark of The X Consortium Inc. Other trademarks belong to the appropriate holders.

Правила оформления документа

В этом разделе перечислены правила, которых следует придерживаться при написании статей для включения в Cyrillic-HOWTO-russian.

По большей части эта информация предназначена для маннейнера, так как документ довольно сильно разросся и приведении всех его частей к единому стилю становится значительной проблемой

Формат

В качестве основного формата документа выбран *XML*. Причиной данного выбора обуславливается историческими причинами. Изначально документ был написан в формате *linuxdoc*. Для преобразования в другие форматы для публикации приходилось пользоваться пакетом sgml-tools. Развитие этого пакета полностью заморожено. Для работы с кириллическими символами требовался напильник. Для корректной работы с *XML* уже почти есть всё необходимое, но опять же пока существует масса трудностей и на текущей стадии возможно корректное преобразование только в *HTML*.

Документ разбит на несколько файлов. Головном файле подключаются главы, каждая из которых лежит в своей директории. Для определения формата в начале каждого файла пишется заголовок

```
<?xml version="1.0" encoding="koi8-r"?>
<!DOCTYPE chapter PUBLIC "-//OASIS//DTD DocBook XML V4.2//EN"
 "http://www.oasis-open.org/docbook/xml/4.2/docbookx.dtd">
```

Сборка для публикации

В качестве базового дистрибутива используется *ALT Linux Master*. Для конвертации *XML* в форматы публикации используется docbook-style-xsl из „Сизифуса“ от [AltLinux](#).

Сборка осуществляется через Makefile. По команде make dist выполняется сборка дистрибутива для дальнейшего распространения. Makefile написан в предположении, что стили находятся в директории /usr/share/xml/docbook/xsl-stylesheets.

Редактирование

Файл редактируется с помощью редактора emacs (раздел [“Emacs и XEmacs”](#)). Для облегчения редактирования используется пакет psgml. Для быстрого разбора DTD используется compiled DTD— Cyrillic-HOWTO-russian.ced, который получается с помощью команды M^x sgml-save-dtd. Для того чтобы указать emacs, чтобы использовать этот файл, вместо разбора стандартного DTD, в конце каждого файла следует добавить

```
<!-- Keep this comment at the end of the file
Local variables:
mode: xml
sgml-omittag:nil
sgml-shorttag:nil
sgml-namecase-general:nil
sgml-general-insert-case:lower
sgml-minimize-attributes:nil
sgml-always-quote-attributes:t
sgml-indent-step:2
sgml-indent-data:nil
sgml-parent-document:nil
sgml-exposed-tags:nil
sgml-local-catalogs:nil
sgml-local-ecat-files:nil
sgml-default-dtd-file:"..../Cyrillic-HOWTO-russian.ced"
sgml-recompile-out-of-date-cdtd:t
End:
-->
```

Переменная sgml-default-dtd-file указывает на сохранённый *DTD*, отличная от nil переменная sgml-recompile-out-of-date-cdtd показывает, что в случае несоответствия DTD установленному в системе его следует пересобрать. Для проверки валидности документа можно воспользоваться встроенной командой emacs: C^c C^v. XML обязан пройти эту проверку прежде чем его можно преобразовывать в какой-либо другой формат.

Предметный указатель

Первый уровень *primary* должен начинаться с большой буквы, все последующие с маленькой.

Метки

Во всём документе метки должны быть уникальными. Метки для подразделов можно создавать, добавляя окончание к метке раздела. Например метка раздела уровня sect1 id="intro", метка подраздела может иметь вид id="intro-doc".

Что нового

Перечислены только изменения стабильной версии.

Версия 1.0 17 августа 2002

Глобальная реструктуризация: кое-что удалено, кое-что подправлено. Косметическая и смысловая правка—не более.

Версия 1.1 25 августа 2002

Перевод документа на другой базовый формат (linuxdoc похоже скончался или близок к этому состоянию). Борис Тоботрас перевёл linuxdoc в docbook с помощью linuxdoc2docbook из своего пакета TEItools, я же занялся изучением docbook и правкой получившегося чуда. Ну как тормозит Emacs—это нечто. Ну почему HOWTO документы не на LaTeХe набирают—уму не постижимо. Пользуюсь примерами от Alt Linux Team Document Project, читаю книжку "DocBook: The Definitive guide".

Версия 1.1.1 3 сентября 2002

Дмитрий Русинов поправил текст для Samba 3.

Версия 1.1.2 7 сентября 2002

Разбил документ на части, так как Emacs прямо надрывается—без слёз смотреть невозможно (здесь стоит грустный смайлик).

Добавил "Рекомендации переводов сообщений программ", взятые с русской странички переводов, составленные Дмитрием Сивашенко—по-моему разумно.

Версия 1.1.3 8 сентября 2002

Подумал— подумал и скопировал страничку с русского сайта переводчиков от Дмитрия Сиващенко о том как стать переводчиком. Для полного комплекта надо добавить главу про gettext — следующая итерация. Сменил название с Cyrillic HOWTO (rus) на Cyrillic HOWTO (russian).

Версия 1.2.0 21 сентября 2002

Разобрался с compiled DTD для psgml— модифицировал соответствующим образом исходники. Поработал над Makefile— причесал для своих нужд. Добавил ChangeLog и README.

Версия 1.2.1 21 сентября 2002

Добавил раздел [Глава 15. Кириллизация дистрибутивов.](#)

Версия 1.2.2 21 сентября 2002

Добавил раздел index, пока с кириллицей как всегда, но со временем надеюсь, что всё устаканится.

Версия 1.2.3 21 сентября 2002

Добавил раздел [“Правила оформления документа”](#).

Версия 1.3 28 сентября 2002

Начал разгребать накопившиеся правки.

Версия 1.3.1

Раздел [Библиография](#) отформатировал как библиографию, выглядит неахти как, но какая—никакая структура.

Версия 1.3.2

Пара правок раздел [Глава 2. Кириллица](#)— лучше всё-таки для создания патча использовать исходники. Заодно отформатировал и вычитал ещё раз этот раздел. Добавил ссылки на библиографию. Добавил небольшой раздел [“Монгольский язык”](#) и совсем маленький [“Азиатские языки”](#).

Версия 1.3.3

Правки от Александра Воропаева: местоположение PsCyr— раздел [“PSCYR”](#). Обновил информацию в разделе [“sharatype”](#). Ручное форматирование и вычитка раздела [Глава 14. Кириллические шрифты](#). Переписал раздел [“Шрифты TrueType”](#).

Версия 1.3.4

Ручное форматирование и вычитка раздела [Глава 9. Кириллизация программ](#). Добавил ссылок в раздел [Русские кириллические ссылки](#). Правка индексов в разделе [Глава 5. Локализация и Интернационализация](#). С подачи Александра Васильева описан способ кириллизации xfig в разделе [“xfig”](#).

Версия 1.4 29 сентября 2002

Доделываю, то что не сделал вчера.

Версия 1.4.1

Правки от Алексея Мичурина.

Версия 1.4.2

Ручное форматирование и вычитка раздела [Глава 11. Кириллизация библиотек](#).
Добавил раздел “[cygwin](#)”.

Украинская ссылка от Mykola Lyakhovych.
Поправил раздел “[OpenOffice](#)”.

Глава 2. Кириллица

Считается, что кириллица составляет видоизменение греческого алфавита. Но существуют и другие точки зрения, например, Срезневский доказывал, что Кириллица в той форме, в какой она встречается в древнейших рукописях XI в., а тем более тот кирилловский устав, который обыкновенно относится к IX в., не может считаться видоизменением тогдашнего греческого алфавита, потому что греки во времена св. Кирилла и Мефодия употребляли уже не устав (унциалы), а скоропись. Из этого бы следовало, что св. Кирилл или один из его учеников (Климент), вероятный изобретатель Кириллицы, взял за образец алфавит греческий прежних времён, или же, что Кириллица известна была на славянской почве задолго до принятия христианства.

Кириллический алфавит моложе других известных нам теперь систем. Многие учёные задавали себе вопрос, не существовал ли у Славян алфавит ещё до принятия христианства? На такое предположение наводили некоторые места из древних авторов, а именно слова известного монаха Храбра, жившего около X в. и написавшего сочинение о изобретении славянских письмен.

Более определённые, точные и верные известия о славянской азбуке являются вместе с известиями о св. апостолах Славянства Кирилле и Мефодии, хотя и здесь на каждом шагу встречаем трудности и сомнения. Прежде всего источники не согласны в разрешении вопроса, где и когда святые мужи изобрели славянскую азбуку. В одном месте мы читаем, что они отправились в путь уже с переведёнными некоторыми священными книгами, другие же источники позволяют думать, что святые братья начали перевод только в Болгарии. Зато относительно самого факта изобретения азбуки везде мы видим согласие. Все вообще сходятся в том, что славянскую азбуку изобрёл св. Кирилл, а имя брата его св. Мефодия упоминается там лишь потому, что они вообще везде действовали вместе и что их имена всегда и везде писались оба рядом. И так мы видим что все источники говорят об одной славянской азбуке и вопрос сделался бы совершенно ясен если бы не обстоятельство, что славяне имеют не одну, а две азбуки: так названную кирилловскую и глаголическую, или кириллицу и глаголицу; эти названия новейшего происхождения и потому они не могут бросить на историю вопроса ни малейшего света. В источниках в этом отношении мы находим только весьма скучные известия.

До сих пор не решён вопрос, кто именно был изобретателем глаголицы и кириллицы. Теперь следует вопрос, какой алфавит лежит в основании этих двух славянских азбук. Для кириллицы дело совершенно ясно, по крайней мере, в одной части, а именно, что главным её основанием служит греческий уставный алфавит, которого буквы без существенных изменений повторяются в кириллице. Не так легко решить вопрос относительно тех букв, которых недостаёт в греческом алфавите: о них высказано несколько различных мнений; между прочим надо заметить, что некоторые из этих букв похожи на соответственные буквы глаголицы. Ещё труднее решить вопрос о глаголице, которая имеет начертания, на первый взгляд совсем не похожие ни на какой известный алфавит. Положительно знаем только, что кириллица распространена на востоке в странах, принадлежащих к восточной церкви, глаголица же только на западе у католических славян в Хорватии, Иллирии и когда-то может быть в Чехии. Оттого явилось предание, что изобретателем глаголицы был святой Иероним, который считается апостолом этих стран. Форма начертаний букв этих двух азбук с течением времени несколько изменялась и по этим изменениям иногда можно с довольно большой точностью определить время написания памятника.

После принятия христианства, русские приняли кирилловскую азбуку, которой стали писаться первые появившиеся на русской почве рукописи, как Остромирово Евангелие, Изборники Святослава и др., и азбука без существенных перемен просуществовала до времён Петра Великого.

Пётр, как известно, путешествовал по западной Европе, знал языки и много читал. Поэтому неудивительно, что он больше привык к латинской азбуке, которая, как и многое западноевропейское, казалась ему красивее и лучше отечественного. Притом в 1699 году напечатано было амстердамское издание книги Ильи Копьевича под заглавием “Поверстание кругов небесных”, где в особенности курсив представляет поразительное сходство с теперешней нашей печатью. Затем последовали и другие амстердамские издания того же характера. Это по мнению Грота (“Спорные вопросы русского правописания от Петра Великого доныне”, С.-Петербург, 1876) главным образом поразило Петра и подало ему первую мысль о преобразовании церковной печати для светских изданий.

Пётр велел выплыть в Амстердаме новый шрифт, который был привезён в Россию в 1708 г., но этот шрифт признан был не вполне удобным, и уже с 1710 года начались его изменения. Одна из главнейших реформ того времени последовала 1735 г. по распоряжению Академии Наук; она состояла в следующем: буква

"зело" была изгнана, а на её место введено "земля"; отменены были тоже "кси" и "в" (ижица), хотя впрочем эта последняя снова в скором времени была возвращена; прибавлена была буква "и", и снова введено существовавшее уже раньше в России "з"; наконец тогда постановили одинаково пишущиеся слова, вроде з_амок и зам_ок, отмечать ударением. Наконец в 1758 г. последовали новые перемены; эта реформа была проведена учреждённым при академии "Российским собранием" и все постановления по этому предмету были изданы как обязательное руководство для типографий двумя записками Тауберта и Шумахера. Тогда звук "и" постановили изображать тремя знаками: "и", "ї", "у", а именно: "и" постановили писать перед согласными, "ї" — перед гласными и в иностранных словах, кроме греческих, где на месте о пишется "у". Кроме того тогда ввели новый знак "Ю" для звука, который после стал изображаться посредством "ё"; этот последний знак был придуман и введен в употребление Карамзиным.

С тех пор уже не создавалось перемен в русской азбуке, хотя некоторые учёные старались теоретически изменять её в разных направлениях. Так прежде всего надо обратить внимание на то, что русская азбука, как и вообще всякий алфавит не вполне отвечает требованиям живого языка, в особенности, если желать точно передавать звуки литературного русского языка и — что ещё важнее — всевозможные оттенки русских говоров. Поэтому учёные прибегают к различным облегчающим дело способам, в особенности же к диакритическим, т.е. надстрочным значкам.

Русская азбука мало по малу начинает выступать за пределы русского языка и даже славянских наречий. Западные славяне употребляют латинский, более или менее приложенный к своим языкам, алфавит; русская же азбука употребляется за пределами России ещё в Болгарии, и её ввёл с некоторыми изменениями и в Сербию Вук Степанович Караджич.

Символы и кодировки

Чтобы понимать и печатать символы различных языков, система и программное обеспечение должны уметь отличать их от других символов. То есть каждый уникальный символ должен иметь уникальное представление внутри операционной системы или внутри специфического пакета программ. Такая совокупность всех уникальных символов, которые система способна распознать, называется *кодировкой*.

Стандарт ASCII

Во время создания большинства операционных систем никто не позаботился предусмотреть возможность представления информации в программах на других языках, отличных от английского. Поэтому наиболее популярной кодировкой была (и фактически ей и остаётся) *ASCII* (Американский Стандартный Код для Информационного Обмена).

Стандарт ASCII (или ASCII с 7-ми битами) включает в себя 128 уникальных символов. Они подразделяются на символы, которые ASCII определяет как собственно печатаемые символы, и на так называемые *символы управления*, большая часть которых использовалась в старых протоколах связи. Каждому элементу набора соответствует целочисленный символьный код (0-127). Подмножество печатаемых символов представляют те, которые находят на клавиатуре пишущей машинки, с некоторыми добавлениями. Каждый символ занимает одно из значений, представляемых 7ю битами, тогда как старший разряд использовался для целей управления (то есть для управления передачи в старых системах связи).

ASCII с 7 битами была расширена до ASCII с 8 битами (или *расширенный ASCII*). В этой кодировке диапазон символов соответствует кодам от 0 до 255. Младшие биты (0-127) — старый добрый ASCII, в то время как старший разряд добавляет ещё 128 символов. Так как эта кодировка обратно совместима с ASCII (символ все ещё занимает 8 бит и коды полностью соответствуют старому ASCII), эта кодировка стала широко использоваться.

Стандарт ISO

Стандарт ASCII с 8 битами не определяет содержание верхней половины таблицы кодировки. Поэтому МЕЖДУНАРОДНАЯ ОРГАНИЗАЦИЯ ПО СТАНДАРТИЗАЦИИ (ISO) взяла ответственность по определению семейства стандартов, известных как семейство *ISO 8859-X*. Это семейство представляет собой совокупность 8-ми битных кодировок, где младшая половина каждой кодировки (символы с кодами 0-127) соответствует ASCII, а старшая половина определяет символы для различных языков. Например, следующие кодовые страницы определены для:

- 8859-0 — Новый европейский стандарт (так называемый *Latin 0*)
- 8859-1 — Европа, Латинская Америка (также известный как *Latin 1*)

- 8859-2 — Восточная Европа
- 8859-5 — Кириллица
- 8859-8 — Идиш

В *Latin 1* старшая половина таблицы определяет различные символы, которые не являются частью английского алфавита, но присутствует в различных европейских языках (немецкие umlauts, французские диакритические знаки и т.д.).

Другая популярная реализация расширенного ASCII — это так называемая *кодовая страница IBM* (названная по имени компьютерной компании, которая создала эту кодировку для своих персональных компьютеров). Эта кодировка в старшей половине кодовой таблицы содержит псевдографические символы. Программное обеспечение, которое не делает никаких предположений о символах, использующих 8-бит ASCII-данных (то есть не использует их для каких-то своих внутренних нужд) называется *чисто 8-ми битными* (примером не чисто 8-ми битного программного обеспечения является использование в фиде буквы "Н" в качестве управляющего символа). Некоторые более старые программы, разработанные для ASCII с 7 битами в памяти, а не для чистых 8-ми бит, могут работать неправильно с данными в расширенном коде ASCII. Большинство программ, однако, способны работать с расширенным ASCII по умолчанию или для этого требуется простая переконфигурация. Подобные простейшие программы, в этом документе не рассматриваются.

Для информации о том, как создавать *своё* чисто 8-ми битное программное обеспечение, обратитесь к разделу [“Locale зависимое программирование”](#).

Так как в большинстве систем символы описываются 8-ю битами, то нет никакого способа расширить ASCII ещё больше. Способ создания новых символов в ASCII — это просто создание других расширенных ASCII реализаций. Именно так кириллица и была введена в ASCII.

Стандарт ISO 8859-5 уже упоминался как стандарт для кириллицы. Но поскольку^[2] он был разработан без принятия во внимание реальных процессов, проходящих в СССР, то единственное, что было действительно достигнуто с введением этого стандарта, так это только увеличение беспорядка с кодировками кириллицы. В сообществе Internet очень популярна кодировка *KOI8-R* (раздел [“Стандарт КОИ8”](#)). Кодировка ISO 8859-5 используется большими компаниями, создающими программное обеспечение с возможностью обработки кириллической информации. К подобным программам, как правило, относятся большие базы данных, решения, основанные на базе OpenVMS и т.д.

Сказанное выше относится только к пост-российскому пространству — кодировка ISO 8859-5 является стандартом кириллицы по крайней мере в Македонии и Сербии.

Стандарт CP866

Альтернативная кодировка — это та самая IBM кодовая страница, где все специфические европейские символы в верхней половине кодовой таблицы были заменены на кириллицу, оставляя псевдографические символы нетронутыми. Следовательно, это не портило вид программ, использующих для работы текстовые окна, а также обеспечило символы кириллицы в них. CP866 все ещё жив и чрезвычайно популярен в среде MS-DOS и OS/2^[3]. Кроме этого имена в файловой системе fat, vfat кодируется в этой кодировке.

Стандарт CP1251

Кодовая страница *Microsoft CP1251* это попытка Microsoft придумать новый стандарт для кодировки кириллицы в Windows. Насколько я знаю, это не совместимо с чем-либо ещё. Однако, в некоторых странах, например, в Болгарии и, вероятно, в республике Беларусь, cp1251 является стандартной кодировкой для Linux и для e-mail.

Стандарт КОИ8

Стандарт КОИ8 в отличие от CP866 и CP1251, был разработан довольно давно. Ещё раз вернёмся в далёкое прошлое. Обычно все работали только со стандартным (7-ми битным) ASCII. 8-ой бит каждого символа указывал на то, что он или управляющий символ, или просто мусор. Обычно никто не слал данные чисто 8-ми битными (каждый норовил оттяпать 8-ой бит у символов). Разработчики КОИ8 применили очень продуманный подход. Они поместили символы русской кириллицы в верхней части расширенной ASCII таблицы таким образом, что позиции кириллических символов соответствуют их фонетическим аналогам в английском алфавите в нижней части таблицы. Это означает, что если в тексте,

написанном в *KOI8*, убирая восьмой бит каждого символа, *то мы все ещё имеем "читабельный" текст, хотя он и написан английскими символами!*

Следует отметить, что *KOI8-R* подходит только для русских текстов, и как следствие был создан украинский вариант *KOI8: KOI8-U*.

Кроме Русской кириллицы *KOI8-R* описывает ещё и Болгарскую кириллицу. Но кодировка *KOI8* неизвестна и поэтому её употребление в Болгарии бессмысленно, хотя самые первые кириллизации персональных компьютеров были сделаны именно в Болгарии с применением *KOI8*. В настоящее время в Болгарии используется кодировка *CP1251*.

Не удивительно, что *KOI8-R* быстро стал фактически стандартом для русской кириллицы в Internet. [Андрей А. Чернов](#) проделал огромный объем работы, чтобы создать стандарт. Он - автор [RFC 1489](#) ("Registration of a Cyrillic Character Set").

Стандарт UNICODE

UNICODE — частичная реализация стандарта *ISO 10646*, совместим снизу вверх, т.е. первые 256 символов *UNICODE* полностью копируют Latin-1 (*ISO 8859-1*). В настоящее время распределено около 40.000 позиций из возможных 65.535 (2 байта на букву), и им соответственно присвоены стандартные имена. Последнее значительное изменение — введение символа валюты EURO — произошло в сентябре 1998 г. Позиции зарезервированы за буквами практически всех известных алфавитов, включая древнеегипетские иероглифы — то есть, используя этот стандарт, можно писать одновременно на русском и греческом, делая вставки на японском, с использованием одного-единственного шрифта (подробности — `man 7 utf-8`). Позиции, под которые по стандарту надо отводить по 4 байта, то есть 2147418112 символов — пока не заняты.

Подробнее про этот формат можно посмотреть по адресу <http://www.unicode.org>.

Посмотреть набор символов *UNICODE* можно здесь: <http://charts.unicode.org/>.

Языковая специфика

Следует различать "славянские" языки и "кириллическую" письменность.

Из всей группы славянских языков, кириллическую письменность используют русский (ru), украинский (uk), белорусский (be), болгарский (bg). Другие: сербский (si) и хорватский (hr) языки используют латиницу и кириллицу, с преобладанием в последнее время латиницы.

Остальные славянские языки используют модифицированную латиницу с диакритиками: чешский (cs), польский (pl), словацкий (sk).

В советское время (примерно в 1930 годы) Академия Наук СССР разработала около десятка письменностей для неславянских народов республик СССР и братских стран на основе кириллической графики.

Из того, что знаю: монгольский, коми-зарянский, коми-пермяцкий, чувашский, башкирский, чеченский, якутский и т.д. (попробую собрать полный список). Модифицированная кириллица. Большинство букв есть в кириллической секции *UNICODE*.

У многих из этих языков ранее (до 1930 г.) были собственные письменности на основе арабского или латиницы, так что сейчас активно происходят процессы возврата на старую графику. Для тюркских народов более характерен переход на модифицированную латиницу по примеру Турции, нежели на арабскую письменность^[4]

На сайте www.peoples.org.ru собрано подробная информация о языках имеющих хождение на территории России.

Беларуский язык

В качестве стандартной кириллической кодировки используется кодировка *CP1251*.

Для более подробной информации следует обратиться к страничке [Все о беларусизации Линукса](#).

Болгарский язык

В качестве стандартной кириллической кодировки используется кодировка *CP1251*. Вместо кодовой страницы *CP866* используется кодировка *МИК*.

Для получения более подробной информации следует обратиться к источникам, собранным в разделе [Болгарские кириллические ссылки](#).

Македонский язык

В качестве стандартной кириллической кодировки используется кодировка *ISO 8859-5*.

Для получения более подробной информации следует обратиться к источникам, собранным в разделе [Македонские кириллические ссылки](#).

Украинский язык

Специфика кириллизации для Украины состоит в том, что часто нужно использовать украинский и русский одновременно. Отказаться от последнего практически не возможно. Из-за этого макросы Злобина (См. раздел [“Cyrillic console tools”](#)), мало пригодны т.к. не позволяют использовать 3 языка одновременно. Приходится редактировать существующие раскладки. На пример раскладки для иксов можно глянуть на www.linux.org.ua

Украинский алфавит можно посмотреть, есть на cad.ntu-kpi.kiev.ua или на [Странице Верховного Совета Украины](#).

Кроме стандартных *KOI8-U* и *CP1251* на Украине имеет хождение кодировка *RUSCI* или *PCT2018-91* или *CP1125*. Подробнее можно посмотреть здесь: www.cad.ntu-kpi.kiev.ua Эта кодировка используется очень часто, например, в эхе *kiev.money* можно получить + за использование другой.

Кроме *KOI8-U* (*RFC2319*) часто встречаются упоминания о *KOI8-RU*.

Для получения более подробной информации следует обратиться к источникам, собранным в разделе [Украинские кириллические ссылки](#).

Русский язык

Русский алфавит — последовательный ряд букв, передающих звуковой состав русской речи и создающих письменную и печатную форму для русского языка. Русский алфавит восходит к кириллице, в современном виде существует с 1918. Содержит 33 буквы, 20 из них передают согласные звуки (б, п, в, ф, д, т, з, с, ж, ш, ч, ц, щ, г, к, х, м, н, р, л); 10 — гласные звуки (а, э, о, ы, и, у) или (в определённых позициях) сочетания "j + гласный" (я, е, ё, ю); буква "й" передаёт "и неслоговое" или "j"; "ъ" и "ъ" не обозначают отдельных звуков. Русский алфавит служит для основы некоторых других языков.

Пара слов в поддержку буквы "ё". Эту букву незаслуженно забывают при наборе текстов. Более того, некоторые деятели ратуют за её полное упразднение. Однако, коль уж вам довелось узнать русский язык, то говорить и писать на нём следует правильно.

Поставь букву "ё" на её место!

Для проверки правописания надо использовать [“Словарь Александра Лебедева”](#), где есть реальная поддержка "ё".

Для получения более подробной информации следует обратиться к источникам, собранным в разделе [Русские кириллические ссылки](#).

Сербский язык

В качестве стандартной кириллической кодировки используется кодировка *ISO 8859-5*.

Для получения более подробной информации следует обратиться к источникам, собранным в разделе [Сербские ссылки](#).

Монгольский язык

Монгольская письменность на основе кириллицы была разработана в САН во времена СССР. Кроме кириллической в Монголии существует и традиционная письменность, но в отличии от кириллической она не описывается с помощью восьмибитных кодовых таблиц.

Для получения более подробной информации следует обратиться к источникам, собранным в разделе [Монгольские кириллические ссылки](#).

Азиатские языки

Для получения более подробной информации следует обратиться к источникам, собранным в разделе [Азиатские кириллические ссылки](#).

^[2] Это часто случается со стандартами.

^[3] Система MS-DOS жива, OS/2 тоже, несмотря на то, что примерно раз в пол года разносится слух о её смерти :).

^[4] Ярким примером перехода на латиницу является Татария. Совершенно бессмысленная акция при условии, что Татария остаётся в составе России

Глава 3. Консоль

Кириллизация текстового режима — самый простой способ отображать и вводить символы кириллицы. Ниже я опишу способ, как "справиться" с драйвером Linux консоли. Поэтому, если вы пытаетесь кириллизовать консоль в другой системе, то не ожидайте, что описанные приёмы будут в ней работать. Взамен проконсультируйтесь с руководством по драйверу терминала и пошлите мне любую информацию, которую вы найдёте. В этом случае я смогу включить её в дальнейшие версии этого документа.

Консольный драйвер Linux — прекрасный образец хорошо настраиваемого программного обеспечения. Он может менять как шрифты, так и раскладки клавиатуры.

Чтобы настроить консоль, вы можете воспользоваться одним из двух описанных ниже пакетов: [console-tools](#) (См. раздел "[consoletools](#)") или [kbd](#) (См. раздел "[kbd](#)").

В ранних версиях Linux альтернативы [kbd](#) не было, но в свежих дистрибутивах вместо [kbd](#) следует использовать пакет [console-tools](#). Этот пакет кардинально отличается от своего предшественника — пакета [kbd](#) (например, в нем нет команды `setfont` — для совместимости используется эмулирующий скрипт).

Чтобы проверить, какой из двух пакетов присутствует в вашей системе, следует попробовать выполнить команду `consolechars`. Если она успешно выполнилась — тогда у вас есть пакет [console-tools](#); если же нет — у вас скорее всего имеется команда `setfont` и значит, установлен пакет [kbd](#).

Вам будет полезно знать общую схему работы консольного драйвера. Когда некоторая пользовательская программа запрашивает консольный драйвер вывести на экран символ, имеющий код "A", то консольный драйвер сначала ищет код "A" в таблице ACM (*Application Charset Map*). Из таблицы он узнает, какой код "B" согласно кодировке Unicode соответствует коду "A". Вы определяете используемую вами кодировку посредством таблицы ACM. Далее консольный драйвер ищет код "B" в таблице SFM *Screen Font Map*.

Таблица SFM определяет кодировку используемого шрифта подобно тому, как ACM определяет вашу кодировку. От SFM консольный драйвер узнает, какой символ активного шрифта имеет код "B" согласно Unicode и выводит его на экран.

Существуют две важные модификации этой схемы.

Аппаратные ограничения современных компьютеров не позволяют использовать шрифты, имеющие более 512 символов. Следовательно, что иногда консольный драйвер не сможет найти код "B" в SFM. Тогда ему на помощь приходит так называемая *fallback таблица*. Эта таблица определяет для кода "B" возможные его аппроксимации "B1", "B2", ... Например, если "B" является код символа "левая двойная угловая кавычка", то возможно "B1" будет кодом символа "левая одинарная угловая кавычка", а "B2" будет просто кодом символа "<". Другим часто используемым применением fallback таблицы является аппроксимация псевдографических символов посредством других. Например, если используемый шрифт не имеет псевдографических символов, и вы заставите консольный драйвер использовать таблицу `graph.fallback`, то он начнёт чертить окна подобно следующей картинке:

```
| Главное меню |
+-----+
| 1. текст   |
| 2. графика |
+-----+
```

На экране это выглядит не так уж и плохо.

Второй важной модификацией общей схемы работы консольного драйвера является *режим Unicode*.

Описанная выше схема относится к 8ми битовому режиму работы драйвера. Когда же он находится в режиме Unicode, то он ожидает, что пользовательские программы используют Unicode и ожидает получать от них коды символов согласно Unicode. Поэтому в этом режиме ACM не используется. Вы всегда можете переключать консольный драйвер из 8ми битового режима в Unicode и обратно посредством команд `unicode_start` и `unicode_stop`.

Возможно, у вас появился следующий вопрос: а зачем нужна вся эта сложная процедура? Что, если я буду использовать шрифты, имеющие используемую мной кодировку? Это возможно, но возникают некоторые осложнения. Вам нужно сказать программам, что псевдографические символы в шрифте уже находятся на других местах. И даже если вы ухитритесь сделать это посредством специального терминального типа (как например `linux-koi8r`), то появится вторая (уже неразрешимая) проблема — из-за аппаратных свойств VGA, горизонтальная псевдографика получится с разрывами, если символы псевдографики в шрифте не расположены на тех же позициях, что и в кодировке `cp437` (расположение псевдографики в `cp866` идентично `cp437`). Поэтому рекомендуется использовать шрифты, имеющие кодировку на основе `cp437` (`cp866`).

Прежде чем приступить к делу, разберёмся с 8-битовым режимом. Не всегда консольный драйвер Linux умел работать с Unicode. Конечно, тогда не существовали также и ACM, и SFM. Вместо них использовалась 8-битовая экранная таблица перекодировки (*screen map*). Это была просто таблица для перевода от одной кодировки (используемой программой) в другую (кодировку шрифта). Например, `koi2alt.trans` переводит коды `KOI8-R` в коды `cp866`. Для совместимости теперь 8-битовый screen map эмулируется с помощью специальной таблицы ACM.

Установка кириллицы с помощью пакетов [console-tools](#) и [kbd](#) обычно состоит из:

1. Настройки экранного шрифта и SFM. Это делается с помощью программы `consolechars` (пакет `console-tools`) или `setfont` и `mapscrn` (пакет `kbd`).
2. Настройки ACM (или 8-битовой экранной таблицы перекодировки) и fallback-таблицы.
3. Загрузки соответствующей раскладки клавиатуры с помощью программы `loadkeys`.

Большинство дистрибутивов позволяет устанавливать кириллицу по умолчанию во время загрузки. Файлы шрифтов обычно находятся в каталогах `/usr/share/consolefonts` или `/usr/lib/kbd/consolefonts`, таблицы ACM, SFM и fallback в `/usr/share/consoletrans`, клавиатурные раскладки в `/usr/share/keymap/i386/qwerty`.

Ниже описано, как установить русскую клавиатурную раскладку. Здесь не используется SFM и fallback таблиц, а вместо ACM используется 8-битовая экранная таблица перекодировки. Это делается в основном из-за того, что у имеющихся в `kbd` и `console-tools` шрифтов для `cp866` нет таблиц SFM. Если вы хотите воспользоваться всеми возможностями консольного драйвера Linux и радоваться хорошим шрифтам, то используйте пакет `console-tools-cyrillic` Виктора Вагнера (См. раздел [“Cyrillic console tools”](#)).

consoletools

Сам пакет можно найти по адресу [console-tools](#).

В том случае, если ваш дистрибутив использует пакет `consoletools`, вам необходимо выполнить следующие действия (естественно, эти команды не следует выполнять дословно):

```
foo="/место/где/лежат/ваши/консольные/шрифты"
loadkeys $foo/ru.map
consolechars -v -f $foo/Cyr_a8x16 -m $foo/koi2alt
```

После выполнения этих команд и загрузки соответствующих файлов вы можете переключать раскладку клавиатуры для ввода кириллических символов с помощью правого *Control* (иногда переключение по умолчанию "вешается" на *Alt* или *Caps Lock*).

К сожалению, шрифты и клавиатурные раскладки, поставляемые вместе с этими пакетами, не позволяют воспользоваться всеми возможностями консольного драйвера. Поэтому я рекомендую вам использовать пакет `console-tools-cyrillic` Виктора Вагнера (См. раздел [“Cyrillic console tools”](#)).

Cyrillic console tools

Этот пакет содержит кириллические консольные шрифты и таблицы перекодировки для Linux, а также раскладки клавиатуры и прочие полезные вещи. Последнюю версию этого пакета можно обнаружить по адресу [ftp://ftp.ice.ru/pub/fonts/linux/](http://ftp.ice.ru/pub/fonts/linux/). Этот пакет активно развивается Виктором Вагнером (vitus@ice.ru). Для настройки следует первым делом следовать прочитать файл `README`, который идёт в поставке. Для работы с пакетом необходимы утилиты `consoletools` (См. раздел [“consoletools”](#)), впрочем, экранные шрифты можно использовать и с последними `kbd`.

Все шрифты сделаны на базе альтернативной кодировки (*CP866*). Почему? Потому что использование любой другой кодировки русских букв в качестве физической в текстовом режиме VGA приводит к тому, что горизонтальная псевдографика получается с разрывами. Это аппаратное свойство VGA, которое никак не лечится. Единственный выход — использовать шрифты, в которых псевдографика расположена на тех же позициях, что и в *CP437*.

В пакет входят семь комплектов шрифтов различной степени пригодности и функциональности. Все шрифты содержат в себе таблицу отображения в *unicode*.

В дистрибутив пакета включены кириллические таблицы перекодировки из распространенных кодировок русского, белорусского, болгарского, сербского и украинского языков в *unicode*. Использование этих таблиц перекодировок совместно со шрифтами, включёнными в этот пакет, позволяет вам настроить консоль в любую требуемую кодировку кириллицы.

В пакет включена комплект для генерации клавиатуры, основанный на идеях, предложенных Кириллом Злобиным (slobin@ice.ru). Раскладки клавиатуры получаются с помощью обработки препроцессором *C* (`/usr/bin/cpp`) файла кодировки и собственно файла раскладки. Использование макросов для описаний клавиш позволяет существенно полнее чем в большинстве распространённых раскладок определить комбинации Alt-клавиша, Ctrl-клавиша и т.д., как в русском, так и в латинском режимах.

Для настройки консоли я использую команды:

```
consolechars -f UniCyr_8x16.psf -m koi8-r.acm
loadkeys console_russian.map
```

Для создания клавиатурной раскладки `console_russian.map` выполняю команду:

```
cpp -DWINONLY -DBACKSPACE cyr-russian-ms.cpp > console_russian.map
```

Это microsoft раскладка, переключение осуществляется с помощью дополнительной клавиши Windows "клавиши системного меню".

P.S. Для украинизации вместо `koi8-r.acm` надо подставить `koi8-u`

kbd

Довольно много дистрибутивов *Linux* устанавливают `kbd` как обязательную часть системы. Сам пакет доступен по адресу [kbd](#).

Пакет `kbd` содержит утилиты управления клавиатурой, кроме этого, с ним поставляется широкий выбор шрифтов и раскладок.

Установка кириллицы с помощью `kbd` обычно состоит из:

1. Загрузки соответствующей раскладки клавиатуры с помощью программы `loadkeys`. Практически в каждом дистрибутиве можно установить раскладку клавиатуры, которую система загружает по умолчанию во время загрузки, с помощью соответствующей программы настройки (например, для RedHat — это `/usr/sbin/kbdconfig`). Вы также можете просто запустить `loadkeys` из вашего `~/.profile` или сделать это руками.
2. Настройки экранного шрифта. Это делается с помощью программы `setfont`. Файлы шрифтов находятся в `/usr/lib/kbd/consolefonts`.

Если вы являетесь приверженцем программ, работающих в текстовом режиме и использующих достоинства псевдографики РС (таких как *Midnight Commander*), то вы можете предпочесть использовать шрифты с CP866 кодировкой и *консольной раскладкой* (*console character map*). Это означает, что ваша консоль отображает CP866 шрифты, но все кириллические символы, соответствующие *KOI8-R* кодировки, соответственным образом отображаются в CP866 и поэтому выглядят правильно. Преимущество этого метода заключается в том, что он позволяет использовать псевдографические символы кодировки CP866.

Ниже перечислены команды, которые позволяют достичь этого эффекта.

```
loadkeys /usr/lib/kbd/keytables/ru.map
setfont /usr/lib/kbd/consolefonts/Cyr_a8x16
mapscrn /usr/lib/kbd/consoletrans/koi2alt
# ниже идёт магическая последовательность
echo -ne "\033(K"
```

При кирилизации всех консолей при загрузке системы вместо строчки
`echo -ne "\033(K"`

следует использовать строчку которая кирилизует все терминалы
`for i in 1 2 3 4 5 6 7; do echo -ne "\033(K" > /dev/tty$; done`

Магическая последовательность необходима для перекодировки вывода символов на экран, если вы используете CP866 шрифты. Она работает, и этого достаточно для спокойной жизни. Однако, если вам любопытно, то посмотрите документацию к пакету `kbd`.

Проверка правильности кирилизации консоли

Теперь вы, вероятно, хотите проверить правильность кирилизации консоли.

Соответствующим образом настройте `bash` (раздел "[bash](#)") или `tcsh` (раздел "[csh/tcsh](#)") — этот шаг *необходим*, перегрузите его, затем нажмите правую клавишу Control. Удовостеритесь, что вы можете правильно печатать по-русски. Клавиша 'q' должна соответствовать "й", 'w' соответствует "ц", и т.д (qwerty->цикен).

По той причине, что на данный момент существует множество самых разных дистрибутивов *Linux*, клавиша переключения в консоли может быть другой. Обычно это правый Alt или Caps Lock — это зависит от раскладки клавиатуры.

Если у вас возникли непредвиденные проблемы, то лучше всего вернуться к родной (то есть US) раскладке. Для этого выполните следующие действия:

```
loadkeys /usr/lib/kbd/keytables/defkeymap.map
setfont /usr/lib/kbd/consolefonts/default8x16
```

ВНИМАНИЕ: В старых версиях *Linux* консольный драйвер не способен сохранить это состояние, когда передаётся управление X Window. Следовательно, после того, как вы вышли из X (или переключаетесь на консоль), вы должны перезагрузить русский шрифт.

Что делать, если "слетела" консоль?

Скорее всего, на консоль был скопирован какой-то двоичный файл, где случайно встретилась переключающая *ESC*-последовательность. Тогда:

Сброс терминала:

```
$ echo -ne "\033c"
```

набрать вслепую, или с работающей консоли:

```
# echo -ne "\033c" >/dev/ttyX
```

Переключение на downloaded font

```
$ echo -ne "\033(K"
```

(возможно, ещё придётся перегрузить шрифты, но это смотря как консоль сорвало...)

Если вывалилась какая-то программа, которая использует *curses*, то проще

```
$ stty sane
```

См. '*man console_codes*' и '*man stty*'

Глава 4. X Window

Подобно консольному режиму, X Window также требует некоторой настройки. Настройка включает в себя настройку ввода и установку шрифтов для X Window. Данные действия обсуждаются ниже.

Установка шрифтов для X Window

Прежде всего, вы должны достать шрифты, содержащие изображения кириллических символов в соответствующих местах. Следует отметить, что все современные дистрибутивы имеют *KOI8-R* шрифты, более-менее пригодные для использования, так что если *xlsfonts* не показывает их, то они просто не установлены или не включены в X Font Path.

X Window включила набор *KOI8-R* шрифтов, разработанных [Cronyx](#), ещё в конце 1995 года. Эти шрифты являются также частью XFree86.

Некоторые дистрибутивы (как правило, очень старые) могут не включать кириллические шрифты для X Window — в этом случае их придётся устанавливать самим.

Подробнее о шрифтах для X Window и о том, где их можно найти, смотрите в разделе "[Шрифты для X Window](#)".

Более современную коллекцию шрифтов для X Window можно взять у [Дмитрия Болховитянова](#) по адресу [пакет CYR_RFX](#).

Вам следует проверить, установлены ли эти шрифты в вашей системе. Спросите системного администратора, или, если вы сами являетесь администратором, проверьте вашу систему сами, а именно:

1. Выполните '*xlsfonts | grep koi8*'. Если в результате выполнения команды появится список шрифтов, то ваш X сервер уже знает об их существовании.
2. Или наберите
 3. *find / -name crox*.pcf**
для того, чтобы найти местоположение шрифтов кириллицы в системе. Вы должны будете сделать эти шрифты доступными для X сервера, как — описано ниже.

Если вы не нашли таких шрифтов в вашей системе, то вы должны установить их сами:

1. Скачайте соответствующий набор шрифтов. Подробнее о шрифтах для X Window и о том где их можно найти можно посмотреть в разделе "[Шрифты для X Window](#)".
2. Теперь, когда у вас есть шрифты, создайте для них каталог. Помещение новых шрифтов в уже существующий каталог шрифтов является, мягко говоря, не очень хорошей идеей. Поместите их, например, в */usr/lib/X11/fonts/cyrillic* для настройки всей системы или просто создайте каталог у себя только для персонального пользования.

3. Если новые шрифты поставляются в формате BDF (файлы *.bdf), то вы должны скомпилировать их. Для каждого шрифта выполните:

4. `bdftopcf -o .pcf .bdf`

Можете сжать их с помощью программы gzip или, если ваш дистрибутив X Window достаточно современный, с помощью bzip2, что даст небольшой выигрыш в степени компрессии:
`gzip *.pcf`

Если же вы все-таки хотите поместить новые шрифты в уже существующий каталог шрифтов, то вы должны "срастить" старый и новый файлы с именем fonts.alias в том случае, естественно, если они оба существуют.

5. В каждом каталоге шрифтов для X должен быть список шрифтов, находящихся в нем. Этот список хранится в файле fonts.dir. Вы не должны создавать этот список вручную. Вместо этого выполните:

6. `cd "директория где лежат шрифты"`
7. `mkfontdir .`

8. Теперь надо сделать этот каталог шрифтов известным для X сервера. Здесь у вас есть ряд возможностей:

- Общесистемная настройка для XFree86. Если вы используете эту версию X Window, то добавьте новый каталог к списку каталогов в файле XF86Config. Чтобы найти его расположение, просмотрите, что скажет startx при запуске (в RedHat этот файл обычно находится в /etc/X11). Более подробносмотрите man XF86Config (4/5).
- Общесистемная настройка через xinit. Добавьте новый каталог к файлу запуска xinit. Более подробно смотрите xinit(1x).
- Персональная настройка. У вас есть специальный файл для запуска X Window — `~/.xinitrc` (им также может быть `~/.Xclients` или `~/.xsessions`)

9. `xset +fp <новый каталог шрифтов>`
10. `xset fp rehash`

Обратите внимание на ключ '+fp' — это означает, что новые шрифты будут добавлены в начале списка каталогов со шрифтами. То есть, если прикладная программа запрашивает, скажем, шрифт fixed, то будет подставлен fixed font с кириллическими символами, чего мы и добивались. Если нужно установить шрифты в нескольких кодировках, то команда xset +fp с предпочтительной кодировкой должна быть после других.

11. Теперь перезапустите X Window. Если вы все сделали правильно, то тесты, описанные в начале раздела, будут пройдены успешно. Также поиграйте с xfontsel (1x), чтобы удостовериться в возможности выбирать шрифты кириллицы.

Чтобы заставить X клиента использовать шрифты кириллицы, вы должны установить соответствующий X ресурс. Например, я делаю кириллический шрифт заданным по умолчанию в моем `~/.Xdefaults`:

`*font: 6x13`

Так как мои шрифты кириллицы идут первыми в списке поиска шрифтов (смотри вывод программы 'xset q'), то они и подставляются, если программа требует какой-либо шрифт с именем, совпадающим с каким-либо из шрифтов, лежащих в кириллической директории.

Вот простые примеры. Если вы хотите научить соответствующий X клиент пользоваться кириллическим шрифтом, то вы должны узнать тип ресурса, который вам надо изменить (это можно сделать, используя editres (1x)) и определить его или в базе данных ресурсов, или в командной строке. Например,

`$ xterm -font 'cronyx-*bold-*-*-*19-*-*-*-*-*'`

запустит xterm с несколько уродливым шрифтом;

`$ xfontsel -xrm '*quitButton.font: -*-times-*-*-*13-*-*-*koi8-*'`

установит Cyrillic Times шрифт для кнопки Quit программы xfontsel.

Использование шрифтов TrueType

Поддержка TrueType встроена во все современные дистрибутивы XFree86. Существует отдельный TrueType сервер шрифтов [XFSFT](#) для X.

Для информации о том, где можно найти TrueType шрифты, следует обратиться к разделу [“Шрифты TrueType”](#).

Коммерческие X сервера (Accelerated X, Metro X) уже довольно давно включили поддержку TrueType в свои дистрибутивы.

Установочная процедура сервера шрифтов довольно проста: делайте то, что сказано в документации.

Немног ниже даётся пример того, как можно установить TrueType шрифты. О том, как подсунуть их X Window — смотрите раздел [“Установка шрифтов для X Window”](#).

Для того, чтобы стандартные шрифты отображались в нужной нам кодировке — в данном случае для примера взята кодировка *cp1251*, — в каталоге, где лежат TrueType шрифты, необходимо создать два одинаковых файла, fonts.dir и fonts.scale, следующего вида:

```
12
timesi.ttf -monotype-Times New Roman-medium-i-normal--0-0-0-0-p-0-microsoft-cp1251
timesbi.ttf -monotype-Times New Roman-bold-i-normal--0-0-0-0-p-0-microsoft-cp1251
timesbd.ttf -monotype-Times New Roman-bold-r-normal--0-0-0-0-p-0-microsoft-cp1251
times.ttf -monotype-Times New Roman-medium-r-normal--0-0-0-0-p-0-microsoft-cp1251
cour.ttf -monotype-Courier New-medium-i-normal--0-0-0-0-m-0-microsoft-cp1251
courbi.ttf -monotype-Courier New-bold-i-normal--0-0-0-0-m-0-microsoft-cp1251
courbd.ttf -monotype-Courier New-bold-r-normal--0-0-0-0-m-0-microsoft-cp1251
cour.ttf -monotype-Courier New-medium-r-normal--0-0-0-0-m-0-microsoft-cp1251
ariali.ttf -monotype-Arial-medium-i-normal--0-0-0-0-p-0-microsoft-cp1251
arialbi.ttf -monotype-Arial-bold-i-normal--0-0-0-0-p-0-microsoft-cp1251
arialbd.ttf -monotype-Arial-bold-r-normal--0-0-0-0-p-0-microsoft-cp1251
arial.ttf -monotype-Arial-medium-r-normal--0-0-0-0-p-0-microsoft-cp1251
```

Кроме этого, сюда же необходимо положить файл encodings.dir, имеющий всего две строки:

```
1
microsoft-cp1251 /usr/X11R6/lib/X11/fonts/encodings/microsoft-cp1251.enc.gz
```

Если вам нужны эти шрифты в кодировке *KOI8-R*, то вместо *microsoft-cp1251* следует прописать *koi8-r*.

Следует отметить ещё одну вещь. Если вы используете сервер шрифтов *xfstt* вместо *xfsft*, то вам следует использовать опцию перекодировки:

```
xfstt ... --encoding koi8-r,windows-1251,iso8859-1
```

После этого новые шрифты будут доступны для X Window — этот факт вы можете проверить, запустив *xfontsel*, и, удостоверившись в наличии семейства шрифтов *ttf*.

Ввод с клавиатуры

В последних выпусках X (X11R61 и выше) имеются два "стандартных" способа ввода с клавиатуры: *Xkb* или *X KeyBoard* (раздел [“Настройка Xkb”](#)) и старый способ, работающий через утилиту *xmodmap* (см. раздел [“xmodmap”](#)).

Настройка Xkb

Полную и исчерпывающую (даже слишком исчерпывающую) документацию по настройке *Xkb* можно найти у [Ивана Паскаля](#).

Немного теории

При старте X-сервера, модуль *XKB* зачитывает все необходимые данные из текстовых файлов, которые образуют "базу данных" настроек *XKB*.

Строго говоря, большинство из этих файлов сам модуль *XKB* не читает. Он вызывает программу *xkbcomp*, которая переводит содержимое этих файлов в двоичный формат, понятный непосредственно модулю *XKB*. Но для настройки это не так уж важно, поскольку вызов *xkbcomp* происходит автоматически, незаметно для пользователя.

База данных, необходимых *XKB*, состоит из 5 компонентов:

- *keycodes* Таблицы, которые просто задают символические имена для скан-кодов. Например:
 - <TLDE>= 49;
 - <AE01> = 10;

- *types* Описывает типы клавиш. Тип клавиши определяет — как должно меняться значение, выдаваемое клавишей в зависимости от модификаторов (*Control*, *Shift* и т.п.). Так, например, "буквенные" клавиши относятся к типу *ALPHABETIC*, что означает, что они имеют разное значение в зависимости от состояния *Shift* и *CapsLock*. А клавиша *[Enter]* имеет тип — *ONE_LEVEL*, что означает, что её значение всегда одно и то же, независимо от состояния модификаторов.
- *compat* (сокращённое от *compatibility*) Описывает "поведение" модификаторов. В XKB имеется несколько внутренних переменных, которые, в конечном счёте, определяют — какой символ будет генерироваться при нажатии клавиши в конкретной ситуации. Так вот, в *compat* как раз описывается — как должны меняться эти переменные при нажатии различных клавиш-модификаторов. В этом же разделе обычно описывается и поведение "лампочек-индикаторов" на клавиатуре.
- *symbols* Таблицы, в которых для каждого скан-кода (имени скан-кода, определённого в *keycodes*) перечисляются все значения (*symbols*), которые должна выдавать клавиша. Естественно, количество различных значений зависит от типа клавиши (которые описываются в *types*), а какое именно значение будет выдано в конкретной ситуации, определяется состоянием модификаторов и их "поведением" (которое описывается в *compat*).
- *geometry* Описывает "геометрию" клавиатуры — то есть расположение клавиш на клавиатуре. Эти описания нужны не столько самому X-серверу, сколько прикладным программам, которые рисуют изображение клавиатуры.

Все эти компоненты разложены по одноимённым директориям в директории `{XROOT}/lib/X11/xkb` (в дальнейшем, я буду обозначать её `{XKBROOT}`). Весь набор компонентов, необходимых для настройки XKB, описывается в файле конфигурации X-сервера в секции *Keyboard*.

Настройка

Чтобы настроить XKB, в файле конфигурации X-сервера надо задать параметры *XkbRules*, *XkbModel*, *XkbLayout* и, если вам нужно что-то не совсем стандартное — *XkbVariant* и *XkbOptions*.

Например,

```
XkbRules      "xfree86"
XkbModel      "pc104"
XkbLayout     "ru"
XkbVariant    "winkeys"
XkbOptions    "grp:shift_toggle.grp_led:scroll"
```

означает, что XKB должен в соответствии с правилами, описанными в файле `{XKBROOT}/rules/xfree86`, выбрать настройки для клавиатуры типа "pc104" (104 кнопки), русского алфавита (английский алфавит будет включён "по умолчанию"). Если у вас "windows" клавиатура (что верно в 99% случаев), то полезно указать *XkbVariant* как "winkeys", в противном случае оставьте эту опцию пустой — "". В самом конце идут дополнительные опции для вашей "раскладки клавиатуры" — "grp:shift_toggle.grp_led:scroll", которые описывают метод — переключения: две клавиши "Shift", при этом загорается лампочка Scroll Lock на клавиатуре.

Несколько практических рекомендаций.

Небольшое отступление — "о клавише-переключателе рус/лат".

Другие варианты смены кодировки:

1. *grp:toggle* — переключение правым Alt;
2. *grp:shift_toggle* — двумя клавишами shift;

3. *grp:ctrl_shift_toggle* — ctrl+shift;
4. *grp:ctrl_alt_toggle* — ctrl+alt;
5. *grp:caps_toggle* — переключение по CapsLock;
6. *grp:lwin_toggle* — переключение по левой "Win" клавише на Windows клавиатуре (для X Window старше 4.2);
7. *grp:rwin_toggle* — переключение по правой "Win" клавише на Windows клавиатуре (для X Window старше 4.2);
8. *grp:menu_toggle* — переключение по клавише "Контекстное меню" на Windows клавиатуре (для X Window старше 4.2);
9. *ctrl:ctrl_ac* — CapsLock;
10. *grp:switch* — переключение по правому Alt только на момент нажатия^[5];

Когда был написан первый вариант этих рекомендаций, сама раскладка "русской" клавиатуры (*symbols/ru*) включала в себя и "переключатель групп" рус/лат, "подвешенный" на клавишу *CapsLock*. С одной стороны, это было удобно — в простейшем случае достаточно было выбрать "русскую раскладку", и вы автоматически получали и клавишу для переключения "на русский". Но, с другой стороны, это было неудобно для тех, кто предпочитает в качестве переключателя рус/лат другую клавишу (или комбинацию клавиш). Конечно, выбрать другой переключатель не составляло труда, но при этом оставался и переключатель на *CapsLock*, что многим не нравилось. Для того, чтобы убрать его, надо было "залезть" в соответствующий файл и вручную подправлять соответствующую раскладку.

В конце концов (начиная с версии 3.3.4) сами разработчики *XFree* убрали этот "переключатель" из "русской раскладки". Но в связи с этим появились и некоторые проблемы — теперь клавишу-переключатель надо явно "заказывать" при конфигурировании *XKB*.

Самый простой способ настройки — использовать программу для автоматической настройки *X Window*. В *XFree86* такая программа называется *XF86Setup*.

При этом "по умолчанию" используются "правила" (*XkbRules*) — *xfree86*. Вам нужно будет только выбрать "модель" (*XkbModel*), "схему" (*XkbLayout*) и "способ переключения групп" (переключатель "РУС/ЛАТ"). Кроме того, при желании вы можете изменить "положение клавиши *Ctrl*". Естественно, в конфигурации это будет выглядеть как соответствующие строчки *XkbOptions*.

Итак. Запустите программу *XF86Setup*, выберите раздел *Keyboard*. В этом разделе выберите из меню *Model* (тип клавиатуры) и *Layout* (язык). Не забудьте отметить в отдельных списках (в правой части) подходящий "переключатель групп" и, если хотите — "расположение *Ctrl*".

При выходе из программы она запишет соответствующие строчки в файл конфигурации *XFree* в секции *Keyboard*.

Для смены кодировки можно воспользоваться любой из многочисленных программ переключения кодовых страниц через *xcb* — например, я пользуюсь программой *xxkb* [Ивана Паскаля](#).

xmodmap

Описанное ниже относится к относительно старым версиям *X Window* (Заведомо к версиям, меньшим 3.3.3), в которых способ *Xkb* (раздел "[Настройка Xkb](#)") не работает.

В случае русификации через *xmodmap* первое, что вы должны сделать, — это *отключить Xkb*!

Чтобы отключить поддержку *Xkb*, в файле XF86Config изучите раздел Keyboard и закомментируйте все строки, начинающиеся с *Xkb* (регистр не имеет значения). Взамен добавьте следующую строку:

XkbDisable

Программа xmodmap позволяет настроить коды, соответствующие различными символам и их комбинациям. Эта настройка основана на файле, содержащем таблицу перекодировки.

Я настоятельно советую вам самостоятельно не разбираться с xmodmap, по крайней мере для начала. Вы не получите ничего, кроме головной боли и разочарования. Вместо этого я рекомендую вам установить пакет [xruskb](#), который позволяет вам сконфигурировать большинство входных параметров перекодировки без необходимости сталкиваться с xmodmap.

Нерусская кириллица для X Window

Все описанное приложимо не только для русского языка. Надо изменить только клавиатурные раскладки, ну и шрифты, естественно, будут иметь другую кодировку.

Возможно, найденные вами шрифты окажутся в отличной от нужной вам кодировке. Для перекодировки BDF-шрифтов можно использовать программу *trbdf* из пакета [trscripts](#).

Эта программа поддерживает кодовые страницы *ibm855*, *ibm866*, *iso-8859-5*, *koi8-c*, *koi8-r*, *koi8-u*, *mac-cyrillic*, *mik*, *unicode* и *esta-cyrillic*. Символы, отсутствующие в исходной кодировке, аппроксимируются, так что её можно использовать, например, для генерации *KOI8-U* шрифтов из *KOI8-R*.

Начиная с версии 4.0.3 XFree официально поддерживает кодировку *cp1251*. Если версия вашей XFree меньше 4.0.3, то поддержку *cp1251* можно получить тремя способами:

1. Использовать дистрибутив, поддерживающий *cp1251*.
2. Патчить *XFree*. [Патч поддержки cp1251 для XFree 3.3.5 и 4.0](#) сделан Алексеем Новодворским.
3. Идеологически неверное решение, но работает везде и легко осуществимо: Добавьте в конце файла *locale.alias* (обычно он находится в каталоге /usr/X11R6/lib/X11/locale) строку вида
4. *be_BY.CP1251 en_US.ISO8859-1*
Конечно, вместо *be_BY.CP1251* здесь Вы можете поставить строку для используемой Вами локали.
(Однако имейте ввиду, что имеющаяся уже в этом файле строка о Болгарской локали *bg_BG* неверна и её нужно удалить.)

Если для кириллизации клавиатуры используете метод через утилиту xmodmap, то этим все и кончится. Если же вы захотите использовать *Xkb* метод, то в клавиатурной раскладке вместо символьических имён нужно использовать коды символов согласно *cp1251*. Например, для белорусской раскладки вместо

```
key <AD01> { [ q, Q ], [ Cyrillic_shorti, Cyrillic_SHORTI ] };
```

нужно использовать

```
key <AD01> { [ 0xe9, 0xc9 ], [ 0xe9, 0xc9 ] };
```

Замечание. Только версии *XFree* после 4.0.0 поддерживают *KOI8-U*. Если у вас отсутствует поддержка *KOI8-U*, а вам нужна именно эта кодировка, то вы можете воспользоваться всем вышеописанным про *cp1251*.

[§] В случае замены *toggle* на *switch* в комбинациях, описанных выше, переключение регистра возникает только на момент нажатия комбинации.

Глава 5. Локализация и Интернационализация

Основной объём этого документа занимают описания, как заставить различные программы понять кириллицу. Обычно каждая программа требует, чтобы это был её собственный метод, как правило, чрезвычайно отличный от других. Кроме того, у некоторых программ поддержка языков, отличных от английского далека от идеала, не говоря уж об их неспособности взаимодействовать, используя родной язык пользователя вместо английского.

Проблемы, перечисленные выше, сильно осложняют жизнь, так как программное обеспечение редко создаётся только для местного рынка. Переработка существенных частей программного обеспечения каждый раз при входе на новый международному рынок очень неэффективна. Интернациональная

поддержка, осуществляемая собственными средствами программы, уникальным и присущим только ей способом, в терминах долгосрочного планирования далеко не блестящая идея.

Следовательно, возникает потребность в стандартизации. *И такой стандарт существует.*

Все связанное с вышеперечисленными проблемами разделено в соответствии с двумя базисными концепциями: *localization* и *internationalization*. Под локализацией мы имеем в виду создание программ, способных обрабатывать различные языковые соглашения для различных стран. Позвольте привести пример. Формат даты, принятый в Соединённых Штатах, имеет вид ММ/ДД/ГГ. Однако в России наиболее популярный формат — ДД.ММ.ГГ. Другие проблемы включают в себя представление времени, форматы числа и представления валюты. Кроме этого, один из наиболее важных аспектов локализации — это определение соответствующих классов символов, то есть определение, какие символы в наборе символов являются "кирпичиками" языка (буквами) и как они упорядочиваются. С другой стороны, локализация не работает со шрифтами.

Интернационализация (или *i18n* для краткости), как предполагается, решает проблемы, связанные со способностью программы взаимодействовать с пользователем на его родном языке.

Обе эти концепции должны быть стандартизованы, давая программистам непротиворечивый путь создания программ, работающих в национальной среде.

Хотя стандартизация ещё в процессе, много её частей уже фактически являются стандартом, так что они могут использоваться без особых проблем.

Я опишу общую схему создания программ, использующих описанные выше возможности стандартным способом. Так как это заслуживает отдельного документа, я буду давать только очень общее описание и указатели на более полные источники.

Locale

Одно из основных понятий локализации — *locale*. Под *locale* подразумевается набор соглашений, специфических для отдельно взятого языка в отдельно взятой стране. В общем случае говорить, что *locale* определяется только страной, неправильно. Например, в Канаде могут быть определены два *locale* — язык "Канада/Английский" и язык "Канада/Французский". Более того, язык "Канада/Английский" не является эквивалентом языку "Великобритания/Английский" или "Американский/Английский", точно так же "Канада/Французский" язык — не эквивалент языку "Франция/Французский" или языку "Швейцария / Французский" .

Более подробное описание проблем/возможностей/достоинств локализации на русском языке можно найти на странице [Локализация, как она есть](#).

Locale с точки зрения пользователя

Каждая *locale* — это специальная база данных, определяющая, по крайней мере, следующие правила и соглашения:

1. Классификация символов и преобразований,
2. Представление валюты,
3. Представление чисел (то есть Десятичные символы),
4. Формат даты / времени.

Настройка локали

Прежде всего — подробная документация о локали имеется на www.sensi.org/~alec/locale Обращайтесь туда, если вам нужны нестандартные варианты (например, отключение русскоязычного интерфейса с сохранением правильной сортировки и т.д.)

Документацию по локали X Window можно найти по адресу www.tsu.ru/~pascal/x_locale/
Вот инструкция для нетерпеливых (только для glibc).

Вам нужно:

- Зайти в /usr/share/locale и создать там симлинк *ru_RU.KOI8-R*, указывающий на *ru_SU*. Эта операция необходима только для *glibc < 2.1.2*.
- Как-либо прописать установку переменной LANG в стартовых скриптах. В RedHat-based дистрибутивах это делается путём редактирования файла /etc/sysconfig/i18n, где, кроме прочего, должна быть строчка
- `LANG=ru_RU.KOI8-R`
В общем случае можно прописать в /etc/profile
`LANG=<ваша кодировка>`
`export LANG`
- Проверить, что все работает, можно, запустив *locale* и посмотрев, что она выдаёт, а также набрав *cal, date* — названия месяцев и дней недели должны быть по-русски.

Гораздо же честнее сделать отдельный *настоящий* каталог: /usr/share/locale/ru_RU.KOI8-R/ (конечно, если его нет в данном дистрибутиве).

Некоторые дистрибутивы *неправильно* включают

```
LANG=ru
LC_ALL=ru_RU.KOI8-R
```

Это *НЕПРАВИЛЬНО*, почему так делать нельзя — описано ниже.

А теперь поговорим о том же, но гораздо подробнее. Итак:

Как включить локализацию?

Если на UNIX машине (с POSIX:1996) средства *locale* правильно установлены и программы правильно написаны, то локализация включается путём задания строки окружения LANG:

```
$ export LANG={язык}
```

Если такой строки окружения нет, то работает значение локализации по умолчанию: LANG="C" или LANG="POSIX" (что то же самое) — минимальный набор параметров, необходимый для функционирования программ на ANSI C (ISO 9899:1990), в кодировке US-ASCII (7 bit) ([“Символы и кодировки”](#)).

Если ваша система имеет полный набор утилит POSIX.2, то узнать установленные в системе и допустимые значения для LANG= можно командой *locale*:

```
$ locale -a
```

По новому стандарту (POSIX.2 приложение E) значения локализации записываются в форме:

```
language_TERRITORY.Codeset
```

или формально:

```
language[_TERRITORY[.Codeset[@modifier]]]
```

Стандарт ISO 639 описывает "language names", ISO 3166 — "territory names". Территории _SU более не существует (вернее, теперь она означает Судан), однако для совместимости некоторые системы продолжают её поддерживать как alias: ru_SU —> ru_RU.

Для русского языка LANG устанавливается, как правило, равным LANG="ru_RU.KOI8-R" или LANG="ru_RU.ISO_8859-5". То есть:

```
$ export LANG="ru_RU.KOI8-R"
```

Для установки украинской локали эта переменная должна быть равна соответственно uk_UA.koi8-u.

Согласно стандарту допустимы также короткие именования значений *locale*, которые часто оформляются как aliases (псевдонимы) полного наименования. Например "C" —> "POSIX".

```
$ export LANG=ru
```

```
$ export LANG=ru_RU
```

```
$ export LANG=ru_RU.KOI8-R
```

Однако, если вы указываете короткое имя, может оказаться, что ваша кодировка оказывается вовсе не KOI8-R (почему следует использовать именно *koi8-r*, описано в разделе [“Символы и кодировки”](#)). Лучше не пользоваться значениями по умолчанию, а указывать точное длинное имя.

Bo FreeBSD 2.x так и есть. Для Linux — зависит от дистрибутива. В коммерческих реализациях (Solaris, SCO, AIX etc), как правило, используется значение LANG="ru_RU", или укороченное LANG="ru" (и, как правило, кодовая страница ISO8859-5 по умолчанию).

Некоторые могут пожелать сделать себе локализацию в другом наборе символов: ru_RX-CP-866 (ru_RU.IBM866), ru_RU.x-mac-cyrillic, ru_RU.ISO_8859-5 или даже ru_RU.CP1251 — на это нет никаких ограничений. Все эти кодировки совершенно равноправны и зарегистрированы (кроме x-mac-cyrillic) в IANA. Только не забудьте, что локализация, ввод-вывод и отображение национальных символов на терминале — это совершенно разные вещи.

Если система локализована не полностью и использовать полное переключение на другой язык (с помощью export LANG={язык}) нельзя, можно включить *locale* только для функций *locale API* библиотеки *libc*, задав значение категорий локализации. Можно также присваивать разные значения разным категориям, задавая их имена в строках окружения:

Если вас раздражают русские даты, сообщения и man-ы, но нужно обрабатывать русские буквы и т.д., то сделайте:

```
$ export LANG="C"
$ export LC_CTYPE="ru_RU.KOI8-R"
$ export LC_COLLATE="ru_RU.KOI8-R"
$ export LC_TIME="C"
```

Ниже идёт описание различных опций *locale*

- LC_CTYPE — определяет одиночные символы,
- LC_NUMERIC — формат чисел,
- LC_TIME — формат времени,
- LC_COLLATE — используется для сравнения строк,
- LC_MONETARY — валюта,
- LC_MESSAGES — системные сообщения,
- LC_PAPER — формат бумаги,
- LC_NAME — формат имён,
- LC_ADDRESS — формат адресов,
- LC_TELEPHONE — формат телефонов.

Не рекомендуется использовать строку окружения:

```
$ export LC_ALL={язык}
```

поскольку формально такой категории локализации нет, она "виртуальная" и обозначает "одновременно все категории". Из-за этого во многих реализациях locale API возникают проблемы. Проблемы могут возникнуть также с программами, работающими с PostScript: в категории LC_NUMERIC локализации ru_RU в соответствии со стандартом ГОСТ в качестве десятичного разделителя используется символ 'запятая': "," в то время, как в стандарте языка PostScript — точка ".". А категория LC_NUMERIC оказывает влияние на printf("%f",float);.

Используйте значение C (POSIX) для LC_NUMERIC, если вы работаете с PostScript:

```
$ export LC_NUMERIC="POSIX"
```

Посмотреть текущие значения категорий локализации можно все той же утилитой *locale* (без параметров).
\$ locale

ПРИМЕЧАНИЕ: В некоторых современных системах начинает появляться локализация в UNICODE.

Например, для России эта локаль включается заданием строки окружения LANG="ru_RU.UTF-8".

Locale зависимое программирование

C locale программа не должна знать о различных символьных преобразованиях и правилах сравнения, описанных выше. Вместо этого они используют специальный API, который действует по правилам, определённым locale. Кроме того, нет необходимости для программы пользоваться только одной locale для соблюдения всех правил — возможно пользоваться другими правилами, описанными в других locale (хотя такой метод не очень хорош).

Из man *setlocale(3)*:

Программа может быть сделана переносимой для всех locale, вызывая setlocale(LC_ALL, "") после инициализации программы, используя значения, возвращённые из запроса localeconv() для locale-зависимой информации, и используя strcoll() или strxfrm() для сравнения строк.

Довольно легко определить четыре уровня программной локализации:

1. Чисто 8ми битное программное обеспечение. То есть программа вызывает setlocale(). Она не делает каких-либо предположений относительно 8-ого бита каждого символа, используя пользовательские функции из ctype.h и ограничения из limits.h, а также заботится относительно signed/unsigned результата. Очень важно, чтобы программа *не* делала каких-либо предположений относительно характера набора символов и их упорядочения. То есть следует воздержаться от следующих конструкций при программировании:
 2. if (c >= 'A' && c <= 'Z') {
 3. ...
- Подобные конструкции с точки зрения правильного "locale"-программирования совершенно недопустимы. Взамен во всех таких случаях должны использоваться макрокоманды из locale зависимого файла заголовка ctype.h. Например:

```
if (isalpha(c) && isupper(c)) { ... или
    if (isascii(c) && isupper(c))
```

Хорошо написанная программа должна быть полностью 8-бит прозрачна. Например, отметка удалённого файла в MS-DOS кодом 0x0E5 — не очень хорошее решение. Ещё примеры плохих решений: знаменитая русская буква "Н" в редакторе GoldEd или русская буква "р" в Norton Commander...
4. Форматы, методы сортировки, размеры листа бумаги. Программа использует strcoll() и strxfrm() вместо strcmp() для строк, использует time(), localtime(), и strftime() для работы со временем, и в заключение, использует localeconv() для правильного представления чисел и валюты.
5. Видимый текст складывается в каталоги сообщений/. Программа должна локализовать весь видимый текст в специальных *каталогах сообщений*. Они содержат соответствия строк на английском и их переводы на другие языки. Выбор сообщений, соответствующих языку окружения, выполнен так, что полностью прозрачен и для программы и для пользователя. Чтобы использовать эти средства, программа должна вызвать gettext() (Sun/POSIX стандарт), или catgets() (X/Open стандарт). Подробнее см. раздел "[Интернационализация](#)".
6. EUC/Unicode поддержка. На этом уровне, программа не использует тип char. Взамен этого она использует wchar_t, который определяет объекты, достаточно большие, чтобы содержать символы Unicode. ANSI C определяет этот тип данных и соответствующий API.

Интернационализация

В то время как локализация описывает как адаптировать программу к иностранному окружению, *интернационализация* (или *i18n* для краткости) детализирует способы общения программы с не-англоговорящим пользователем.

Прежде это делалось с помощью создания абстракций сообщений для вывода их из кода программы. Теперь такой механизм (более или менее) стандартизирован. И, конечно, есть его free реализации!

Проект GNU, наконец, стал на путь создания интернационализированных прикладных программ. Ulrich Drepper (drepper@ipd.info.uni-karlsruhe.de) разработал пакет gettext. Этот пакет лежит во всех GNU архивах, например, в prep.ai.mit.edu. Он позволяет вам разрабатывать программы в направлении, двигаясь в котором вы можете легко заставить их поддерживать большее количество языков. Пакет gettext поставляется с превосходным руководством.

Как принять участие в проекте перевода

Все подробности можно узнать на страничке [Русской команды переводчиков](#). Информацию о переводах программ лучше взять на [официальном сайте проекта](#). Кроме команды перевода GNU существуют и другие локальные проекты по локализации программ и переводу документации. Если вы заинтересованы в кириллизации проекта *GNOOME*, то следует обратиться к [страничке перевода GNOOME](#). Страница команды перевода kde расположена на [официальной странице проекта KDE](#).

Для того, чтобы стать переводчиком сообщений программ в рамках проекта 'Free Translation Project', Вам не обязательно нужно быть программистом или очень квалифицированным пользователем. Гораздо важнее хорошее знание русского языка (в том числе, в области компьютерной терминологии), умение избегать орфографических ошибок и грамотно формулировать фразы.

Ниже приведено подробное пошаговое описание действий, необходимых для эффективного начала работы в качестве переводчика сообщений программ на русский язык.

- *Станьте участником команды перевода.*

Чтобы стать участником русской команды перевода, подпишитесь на список рассылки tt@chg.ru. Для этого пошлите E-mail на адрес <majordomo@chg.ru> указав в теле письма строку:

`subscribe tt <yourname@yourdomain.com>`

В этой строке замените <yourname@yourdomain.com> на свой собственный адрес электронной почты. Для отправки писем в список рассылки используйте адрес <tt@chg.ru>. Вы можете писать в список рассылки как по-русски, так и по-английски.

- *Установите GNU Gettext и научитесь с ним работать.*

Вы могли бы работать над переводами без установки GNU Gettext. Однако, в этом случае Вы не сможете проверить работоспособность перевода. Если Вы пользуетесь текстовым редактором GNU Emacs, Вам будет полезен режим редактирования PO-файлов, который упрощает работу с файлами перевода. Кроме того, GNU Gettext содержит документацию, которая очень детально объясняет процесс перевода.

Вы можете взять дистрибутив GNU Gettext с <ftp://ftp.gnu.org/pub/gnu/> или с любого зеркала архива GNU. Но, возможно, лучше использовать последнюю неофициальную версию с основного архива проекта перевода GNU ([FTP](#), [WWW](#)). Если Вы пользуетесь программой GNU Emacs, возьмите также файл для поддержки режима редактирования PO-файлов ([FTP](#), [WWW](#)).

И, наконец, прочитайте info-документацию пакета GNU Gettext. Можно подготовить и печатную версию документации, выполнив команды "make dvi ; dvips gettext.dvi" в каталоге doc/ дистрибутива Gettext.

- *Заполните и пошлите "disclaimer" для переводчиков.*

Чтобы Ваши переводы были включены в дистрибутивы программ GNU, необходимо отправить в организацию Free Software Foundation документ, чётко определяющий правовой статус Ваших переводов. Форму для заполнения можно найти по этому [адресу](#).

Распечатайте эту форму, ознакомьтесь с содержанием, заполните, поставьте дату и подпись. После этого запечатайте лист в конверт и пошлите его по указанному там адресу. Возможно, пройдет несколько недель, пока письмо дойдет по назначению и будет обработано. Когда это произойдет, Вы получите E-mail с подтверждением, а также сможете найти свое имя в файле [AUTHORS](#) в центральном архиве проекта перевода GNU.

- Возьмите подходящую программу, сообщения которой ещё не переведены.

Подберите программу или пакет, который Вы желали бы перевести. Страйтесь выбирать те программы, с которыми Вы реально работаете. Кроме того, в первую очередь желательно переводить сообщения наиболее распространенных и необходимых большому числу пользователей программ. Информацию о пакетах программ, для которых требуется перевод, можно найти в [базе данных программ русской команды перевода](#). Если в таблице для какого-либо пакета в качестве статуса указано "Перевод отсутствует", Вы можете стать его переводчиком.

Когда Вы остановили свой выбор на конкретном пакете программ, сообщите об этом остальным участникам команды. В таблице базы данных появится Ваше имя и, следовательно, другие переводчики не будут начинать работать над теми же самыми переводами.

Перед началом перевода не забудьте ознакомиться с рекомендациями по переводу сообщений на русский язык в разделе "[Рекомендации по переводу сообщений программ](#)". Возможно, приведенные там соображения помогут Вам в неизбежных вопросах "Как лучше перевести ту или иную фразу?".

- Включайтесь в реальную работу по переводу!

Допустим, Вы решили перевести сообщения пакета "gettext-N.M" (N.M — номер версии). Возьмите в дистрибутиве программы файл шаблона "po/gettext.po" и скопируйте его в рабочий файл "gettext-N.M.ru.po". Шаблон обычно начинается такими строками:

```
# SOME DESCRIPTIVE TITLE.
# Copyright (C) YEAR Free Software Foundation, Inc.
# FIRST AUTHOR <EMAIL@ADDRESS>, YEAR.
#
#, fuzzy
msgid ""
msgstr ""
"Project-Id-Version: PACKAGE VERSION\n"
"PO-Revision-Date: YEAR-MO-DA HO:MI+DIST\n"
"Last-Translator: FULL NAME <EMAIL@ADDRESS>\n"
"Language-Team: LANGUAGE <LL@li.org>\n"
"MIME-Version: 1.0\n"
"Content-Type: text/plain; charset=CHARSET\n"
"Content-Transfer-Encoding: ENCODING\n"
```

Замените все слова в верхнем регистре реальной информацией и удалите строку с текстом "#, fuzzy". Если возникают сомнения в правильности заполнения заголовка, Вы можете посмотреть в качестве примера уже имеющиеся файлы перевода на другие языки, либо русские переводы сообщений других программ.

Заметьте, что последние строки заголовка для переводов на русский язык должны выглядеть так:

```
"Language-Team: Russian <ru@li.org>\n"
"MIME-Version: 1.0\n"
"Content-Type: text/plain; charset=koi8-r\n"
"Content-Transfer-Encoding: 8bit\n"
```

Как следует из заголовка, весь перевод необходимо выполнять в кодировке KOI8-R, являющейся стандартом де-факто для Unix-подобных свободно распространяемых операционных систем.

Оставшаяся часть PO-файла содержит сообщения, подлежащие переводу. Например, непереведенное сообщение может выглядеть так:

```
#: lib/error.c:88
msgid "Unknown system error"
msgstr ""
```

В пустую строку с ключевым словом `msgstr' нужно вписать русский перевод фразы, показанной в строке `msgid':

```
#: lib/error.c:88
msgid "Unknown system error"
msgstr "Неизвестная системная ошибка"
```

Таким же образом переводятся остальные сообщения в этом файле. Заметьте, что сообщения, для которых не указан русский перевод, будут отображаться русифицированной программой в своем английском варианте. Поэтому возможно переводить сообщения программы частично и это не приведет к пропаданию непереведенных сообщений.

После того, как перевод закончен (или Вы желаете проверить корректность перевода части сообщений), РО-файл нужно обработать утилитой msgfmt для получения индексированного двоичного файла перевода. Например, команда

```
msgfmt -o gettext.mo gettext-N.M.ru.po
```

сформирует файл gettext.mo, который готов для использования переведённым Вами пакетом gettext. Если поместить этот файл в каталог сообщений локализации (например, это может быть каталог /usr/share/locale/ru/LC_MESSAGES/) и установить переменную окружения LANGUAGE в значение "ru", то программы пакета gettext будут отображать переведённые сообщения по-русски. Впрочем, работоспособность этого примера может зависеть от используемой операционной системы, версии системных библиотек GNU, корректности их установки и т.д.

Когда, наконец, РО-файл будет готов, Вы можете отправить его русской команде перевода (например, послать анонс в список рассылки tt@chg.ru). Если с переводом все в порядке, можно отправить перевод на адрес translation@iro.umontreal.ca для последующего включения в официальный дистрибутив пакета программ. Включение файла перевода в официальный дистрибутив может быть отклонено, если от Вас не был получен "disclaimer", если кто-либо другой в команде перевода уже был закреплен за переводом данного пакета программ, или в случае официального отказа русской команды перевода принять Вашу работу.

Команда перевода рассматривается в качестве основного авторитета в отношениях с проектом перевода GNU, поэтому важно согласовывать свою работу с остальными участниками команды.

Более подробную информацию о проекте 'Free Translation Project' Вы можете найти на центральном архиве проекта:

Рекомендации по переводу сообщений программ

Максимум внимания при переводе сообщений следует уделять достижению двух главных целей:

- 1.** *Основная задача перевода — обеспечить точную передачу смысла оригинальной фразы.* При этом единицей перевода является сообщение целиком, а не отдельное его слово или словосочетание. Более того, на перевод фразы может влиять и контекст, в котором она появляется.
- 2.** *Перевод должен обеспечить комфортное восприятие сообщений максимальным количеством потенциальных пользователей программы.* При этом недостаточно, чтобы все понимали смысл сообщений, важно не вызвать у пользователя отторжение из-за наличия грамматических ошибок или неудачного перевода терминов.

Ниже сформулированы несколько важных рекомендаций, которые являются следствиями соображений, приведенных выше.

- *Не нужно переводить сообщение дословно.* Перевод должен отражать суть фразы, а не форму ее выражения на языке оригинала. Типичной ошибкой является старание сохранить неизменными общую форму предложения или даже порядок отдельных слов. Стремление переводить одно английское слово одним и тем же русским эквивалентом также не всегда оправдано.

Пример:

Исходная фраза:

"The following options can be used to define the colors to use for different things:"

Вот вариант дословного перевода, который выглядит совершенно нелепо:

"Следующие опции могут быть использованы для того, чтобы определить цвета, которые следует использовать для различных вещей:"

Более удачный перевод:

"Для указания цвета различных элементов оформления можно использовать следующие ключи:"

- Самое очевидное правило — *не допускать орфографических и грубых фразеологических ошибок*. Как ни парадоксально, в нашем случае для переводчика важнее хорошо знать русский язык, нежели в совершенстве владеть английским. Это объясняется тем, что потреблять результаты перевода будут главным образом те люди, для которых родной язык — русский.
- *Не стоит использовать в переводе разговорный стиль*. Если, к примеру, в письмах электронной почты неформальное выражение мыслей воспринимается вполне естественно, то в сообщениях программ выглядит не всегда уместно. Не нужно придумывать новых "русских" терминов или использовать жаргонные слова (даже широко распространенные в среде людей, профессионально связанных с компьютерами).
- *Не рекомендуется составлять фразы от первого лица*. Такая практика широко распространена в англоязычных сообщениях, но в русскоязычных программах применяется редко.

Таблица 5.1. Пример перевода:

фраза	правильно	не удачно
Can't open file	Ошибка открытия файла	Не могу открыть файл

- *Не рекомендуется начинать каждое слово надписей с заглавных букв*, как это часто делается в англоязычных программах для пунктов меню, подписей к элементам управления окон диалога, заголовков окон. То же относится к заголовкам разделов документации: в русском языке принято делать заглавной только первую букву заголовка.
- *Не следует переводить названия программ, имена людей и общепринятые английские сокращения*.

Главная проблема перевода — сложность подбора русских эквивалентов к некоторым англоязычным терминам: не всегда существует общепринятый русский перевод, а дословная интерпретация часто не соответствует смыслу оригинального термина. В таких случаях трудно что-либо рекомендовать и выбор оптимального варианта остаётся за переводчиком. Неплохая мысль в таких случаях — ознакомиться с другими вариантами перевода данного термина в литературе или локализованных версиях других программ. В качестве крайнего случая, когда подобранный русский эквивалент вызывает двусмысленность, можно оставить в скобках оригинальный термин на английском языке, хотя в идеале таких ситуаций возникать не должно.

Приведённые здесь правила носят рекомендательный характер и не являются "официальной догмой". Их цель — объединить переводчиков единым подходом к работе, в результате которой конечные пользователи получили бы качественно локализованные версии программных продуктов. Любые мнения и пожелания по содержанию этой странички будут с благодарностью приняты.

Глава 6. Редактирование текста

В этом разделе описывается настройка различных программ для работы с кириллическими текстами. Этот раздел не охватывает *текстовые процессоры*, настройка которых описывается позже (смотри раздел [Глава 7. Текстовые процессоры](#))

Проверка правописания на русском языке

Для проверки правописания обычно используется программа GNU ispell. Она имеет очень гибкие настройки и возможности для расширения. Её можно использовать при проверке правописания текстов, написанных на языках, отличных от английского, путём добавления новых *словарей*.

В идеальном случае, когда ваш ispell установлен правильно, вам надо только создать словарь, используя команды, обеспечиваемые файлом Makefile из дистрибутива соответствующего словаря. Однако с некой долей вероятности у вас возникнут проблемы с ispell, который окажется понимать 8-ми битные данные. Это означает, что в вашей системе ispell скомпилирован без поддержки 8-ми битных символов. В этом случае вам не удастся избежать перекомпиляции пакета ispell.

Если у вас все нормально, то можете инициировать проверку правописания для русских текстов путём вызова программы ispell с опцией '-d russian'.

Последняя версия ispell имеет номер 3.1.20. При желании пакет можно взять здесь <ftp.cs.ucla.edu> или найти на любом крупном ftp-сервере. При самостоятельной компиляции ispell надо скопировать файл local.h.samp в local.h, закомментировать в нем *NO8BIT* (#undef NO8BIT) и установить правильные пути для BINDIR, LIBDIR и других рабочих директорий.

Словарь Александра Лебедева

Этот словарь построен на основе словаря русского языка для ispell, первоначально составленного Нилом Далтоном (Neal Dalton) в 1992 г. После тщательной проверки и исправления примерно 4000 ошибок в словаре Нила Далтона (в этой части работы использовался орфографический словарь русского языка, выпущенный Институтом русского языка АН СССР), в словарь были добавлены отсутствовавшие в нём правила образования форм существительных, прилагательных, причастий, наречий, изменены правила формирования окончаний глаголов, так что affix-файл можно считать переписанным заново. Одновременно в словарь было добавлено большое число слов.

Словарь постоянно совершенствуется, дополняется и корректируется. Последнюю версию словаря можно найти на сервере: <mch5.chem.msu.su>

Как можно понять из названия раздела, автором этого словаря является [Александр Лебедев](#).

На основе этого словаря Serge Winitzki создал словарь для проверки правописания в текстах, набранных в дореформенной русской орфографии. Предложенное им расширение кодировки koи8 и сами словари можно найти по адресу: <sawsoft.newmail.ru/LS/koi8-extended.html>.

Отличительной чертой данного словаря является то, что начиная с версии 0.99c0 в него включена полноценная поддержка буквы ё (другие словари просто заменяют букву ё на букву е).

Словарь Константина Книжника

Константин Книжник создал хороший русский словарь для ispell. Вы можете найти его на [домашней странице К.Книжника](#).

Emacs и XEmacs

Есть две версии редактора *Emacs* — GNU Emacs и XEmacs. Они обеспечивают более или менее сходный набор функций и возможностей, но при этом реализация некоторых деталей расходится довольно сильно.

Для настройки старых версий *emacs* следует посмотреть изначальный "Cyrillic HOWTO" от Беликова — дословный перевод можно найти [на моей домашней странице](#) в разделе Cyrillic HOWTO (old rus).

Существует масса способов кириллизации *emacs* — в основном эти способы требуют некоего низкоуровневого (в духе emacs Лиспа) "хака". Более "идеологически" правильным способом кириллизации emacs'a является настройка с использованием MULE (MULtilanguage emacs support) — раздел "[Emacs и MULE](#)".

Совет: При работе с *X Window* для *emacs*, вероятно, лучше всего подойдут шрифты lucidatypewriter из коллекции шрифтов Болховитянова *Cyr-RFX* (См. раздел "[Сур-RFX](#)").

Emacs и MULE

MULE — "MULtilanguage Emacs support" — это подсистема *emacs*, которая создана для возможности использования в этом редакторе различных языков. Основным толчком для развития *MULE* послужило желание использовать китайские/японские и тому подобные языки — с этим, говорят, проблем нет, а вот по поводу использования кириллицы мало что известно — попробуем разобраться.

Для правильной работы с *Mule* надо всего-то

1. Пакет *emacs-leim*. Этот пакет присутствует в подавляющем большинстве известных дистрибутивов.

2. Русские шрифты в *koi8-r* или *iso5* кодировке.

Далее:

1. Идём в меню Mule, Set Font/Fontset, Fontset и выбираем standard. Зачем? Fontset — это набор шрифтов, который *emacs* подбирает себе при загрузке. Причём startup подбирается с тем расчётом чтобы для каждого шрифта из Fontset'a имелись italic и bold варианты, а standard подбирается так, чтобы в Fontset'e имелось как можно больше различных кодировок. Чтобы сделать изменение постоянным, запишите следующую строчку в *~/.emacs*.

2. (set-default-font "fontset-standard")

3. Идём в Mule, Set Language Environment, Cyrillic и устанавливаем один из 3-х вариантов. Какой — по большому счету без разницы. Это будет влиять только на общение *emacs*'а с внешним миром — если захотим Cyrillic-ISO, то текст будет сохраняться в *ISO* и в буфер обмена будет копироваться текст в кодировке *ISO*. То же самое с *KOI8*. Чтобы сделать изменение постоянным, запишите в *~/.emacs*.

4. (set-language-environment "cyrillic-koi8")

Всё. Теперь самое главное — отучиться пользоваться системным переключателем раскладки клавиатуры — *emacs* все делает сам. По умолчанию переключатель раскладки клавиатуры повешен на С-\ (Как не странно — это довольно удобно).

Будьте готовы к тому, что могут выбраться совершенно неязычные шрифты. Чтобы кастомизировать выбор шрифтов, нужно создать свой собственный набор шрифтов (на сленге *emacs* — fontset). Если у вас уже установлены *Cyr-RFX* (См. раздел [“Cyr-RFX”](#)):

1. Наберите что-то вроде

2. (create-fontset-from-fontset-spec
3. " -b&h-* -medium-r-* -14-*-*-* -fontset-vasja,
4. latin-iso8859-1 :b&h-* -medium-r-* -14-*-*-* -iso8859-1,
5. latin-iso8859-15 :b&h-* -medium-r-* -14-*-*-* -iso8859-15,
6. cyrillic-iso8859-5 :b&h-* -medium-r-* -14-*-*-* -iso8859-5" t)

7. Поместите курсор за закрывающей скобкой и нажмите С-х С-е выполнив введённый ранее кусок кода.

8. Затем наберите

9. (set-default-font "fontset-vasja")
и опять-таки выполните этот код, нажав комбинацию клавиш С-х С-е (не забудьте, что курсор должен находиться за последней закрывающей скобкой). Результат должен быть виден сразу же.

10. Добавьте эти команды в *~/.emacs*, чтобы они выполнялись при каждом запуске *emacs*'а.

11. Да, ещё туда же можно добавить

12. (set-language-environment "Cyrillic-KOI8")

Чтобы *emacs* не плевался в консоли что не может найти иксовых шрифтов, хорошо бы обрамить вышеприведённый кусок кода следующим условием:

```
(defmacro Xlaunch (&rest x)
  (list 'if (eq window-system 'x) (cons 'progn x)))
(Xlaunch
  ...
  ...
  ...)
```

Сторонникам теории множественности кодировок (чем больше кодировок — тем лучше): для того, чтобы включить поддержку 1251, вставьте (codepage-setup 1251)

в `~/.emacs`. 866-ю кодовую страницу можно добавить туда же следующим образом:

```
(defvar cp866-decode-table
  [
  255 240 nil nil 242 nil nil 244 nil nil nil nil nil 246 nil
  128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143
  144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159
  160 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175
  224 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239
  252 241 nil nil 243 nil nil 245 nil nil nil nil nil 247 nil]
  "Table for converting ISO-8859-5 characters into codepage 866 glyphs.")
(setplist 'cp866-decode-table
  '(charset cyrillic-iso8859-5 language "Cyrillic-ISO" offset 160))
```

(codepage-setup 866)
Чтобы открыть файл в какой-то кодировке, отличной от дефолтной, идём в Mule, SetCodingSystem, NextCommand, выбираем кодировку и выполняем свою команду (например, C-x-f, открыть файл) То же самое чтобы сохранить - Mule, SetCodingSystem, NextCommand, выбираем кодировку и сохраняем в данной кодировке(C-x-s)

Вот пример готового `/emacs` для шрифтов *Cyr-RFX* — раздел “[“Cyr-RFX”](#)”:

```

;; Вот пример готового .emacs для шрифтов Cyrillic-X
;; только в X Window
(defmacro Xlaunch (&rest x)
  (list 'if (eq window-system 'x)(cons 'progn x)))
;; проверить на наличие иков
(Xlaunch
;; зарегистрировать набор шрифтов
(create-fontset-from-fontset-spec
  "-b&h-*_medium-r-*_*-_14_*_*_*-*_*-fontset-vasja,
  latin-iso8859-1:-b&h-*_medium-r-*_*-_14_*_*_*-*_*-iso8859-1,
  latin-iso8859-15:-b&h-*_medium-r-*_*-_14_*_*_*-*_*-iso8859-15,
  cyrillic-iso8859-5:-b&h-*_medium-r-*_*-_14_*_*_*-*_*-iso8859-5" t)
;; установить набор шрифтов
(set-default-font "fontset-vasja")
)

```

```
;; делать всё в кодировке KOI8-R  
(set-language-environment 'Cyrillic-KOI8)  
  
;; вспомни "виндозную" кодировку...  
;; (1251)
```

;; вспомни "виндозную" кодировку...
(codepage-setup 1251)

```
;; ...и научись "досовской"
(defvar cp866-decode-table
  [
  255 240 nil nil 242 nil 244 nil nil nil nil nil 246 nil
  128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143
  144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159
  160 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175
  224 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239
  252 241 nil nil 243 nil nil 245 nil nil nil nil nil 247 nil]
  "Table for converting ISO-8859-5 characters into codepage 866 glyphs."
  (setplist 'cp866-decode-table
    '(charset cyrillic-iso8859-5 language "Cyrillic-ISO" offset 160))
  (codepage-setup 866))
```

Если вдруг кому-то придёт в голову работать в *emacs* из консоли, просто запустите его с опцией —unibyte. В таком случае работа с *emacs* не будет отличаться от работы в других консольных редакторах.

Добавление: Кусок моего `.emacs`, отвечающего за кириллизацию, выглядит следующим образом (у меня установлен пакет `Cyr-REX` — раздел “`Cyr-REX`”):

```
; установлена пакет "Утилиты" - раздел Справка).  
;;  
;; шрифты для X Window  
;;  
;; Подстановка шрифтов  
(cond (window-system  
      (create-fontset-from-fontset-spec  
        "-rfix-courier-medium-r-*-*-18-*-*-*-fontset-Evgueni" t)  
      ;; по умолчанию  
      (set-default-font "fontset-Evgueni")  
      ;; по умолчанию для фреймов  
      (add-to-list 'default-frame-alist '(font . "fontset-Evgueni"))  
    )  
)  
;;  
;; русификация через mule - переключалка "Ctrl+\" - требуется
```

```

;;
;;(set-language-environment "Cyrillic-KOI8");так
;;(set-language-environment "Cyrillic-koi8-r");или сяк

;; This could help for transferring data from bad old apps
(set-selection-coding-system 'koi8-r)

;;
;; Чтобы раскладка соответствовала раскладке русских букв на виндовых (а теперь
;; таких большинство) клавиатурах, можно добавить еще следующие строки:
;;
;; redefine some orginal keybindings под MS - можно ли попроще?
;;
;; кнопка \| => √ при переключения регистра (для моего удобства)
;;

(quail-define-package
  "cyrillic-jecken" "Cyrillic" "RU" nil
  "ЙЦУКЕН keyboard layout widely used in Russia (ISO 8859-5 encoding)"
  nil t t t nil nil nil nil t)

(quail-define-rules
 ("1" ?1) ("2" ?2) ("3" ?3) ("4" ?4) ("5" ?5) ("6" ?6) ("7" ?7) ("8" ?8)
 ("9" ?9) ("0" ?0) ("-" ?-) ("=" ?=) ("`" ?è) ("q" ?й) ("w" ?и) ("e" ?у)
 ("r" ?к) ("t" ?е) ("y" ?н) ("u" ?т) ("i" ?ш) ("o" ?щ) ("p" ?з) ("[" ?х)
 ("]" ?ь) ("a" ?ф) ("s" ?ы) ("d" ?в) ("f" ?а) ("g" ?п) ("h" ?р) ("j" ?о)
 ("k" ?л) ("l" ?д) (";" ?к) ("{" ?з) ("\" ?ъ) ("z" ?я) ("x" ?ч) ("c" ?с)
 ("v" ?м) ("b" ?и) ("n" ?т) ("m" ?ь) ("," ?б) ("." ?ю) ("?" ?.) ("!" ?!)
 ("@" ?") ("#" ?#) ("$" ?;) ("%" ?%) ("^" ?;) ("&" ??) ("*" ?*) ("(" ?(
 ")" ?)) ("_" ?_) ("+" ?+) ("~" ?Ё)
 ("Q" ?Й) ("W" ?Ц) ("E" ?У) ("R" ?К) ("T" ?Е) ("Y" ?Н) ("U" ?Г) ("I" ?Ш)
 ("O" ?Щ) ("P" ?З) ("{" ?Х) ("}" ?Ь) ("A" ?Ф) ("S" ?Ы) ("D" ?В) ("F" ?А)
 ("G" ?П) ("H" ?Р) ("J" ?О) ("K" ?Л) ("L" ?Д) (";" ?Ж) ("?" ?Э) ("|" ?/)
 ("Z" ?Я) ("X" ?Ч) ("C" ?С) ("V" ?М) ("B" ?И) ("N" ?Т) ("M" ?Ь) ("<" ?Б)
 (">" ?Ю) ("?" ?,))

```

X-ресурсы

Для настройки интерфейса emacs полезно воспользоваться механизмом X-ресурсов. Для кириллизации меню и для задания геометрии окна emacs в файл можно .Xdefaults прописать следующие строки:

```

Emacs.pane.menubar.font: 8x16
Emacs.pane.menubar.* fontlist: -rxf-courier-medium-r-*-*-18-*-*-*-*-koi8-r
Emacs.geometry: 86x51+0+0

```

Шрифты и геометрию следует подобрать согласно вашим требованиям.

vi

Редактор vi (по крайней мере, его клон vim, присутствующий в большинстве дистрибутивов Linux) знает о существовании 8-ми битных символов. Это даёт вам возможность вводить кириллические буквы. Редактор правильно распознает границы слов.

Насколько я понимаю, vim использует *locale* для преобразования букв из строчных в заглавные и обратно, т.е. на всех встречаенных мной *unix-like* системах vim правильно работал с русским языком после настройки *locale* (Как настроить локаль смотрите раздел "[Настройка локали](#)").

Для того, чтобы постоянно не переключаться в командном режиме между английскими и русскими буквами, необходимо выполнить маппирование русских букв в английские — либо (в старых версиях vim или vi) через команду *map* для каждой буквы, либо (в новых версиях vim) через опцию *langmap*. Последнее делается добавлением в ваш *~/.vimrc*, или в общесистемный *vimrc*, двух строк:

```

set langmap=ж;;
set langmap=ё`й,ç,у,е,к,т,н,...<и так далее>...,ь,м,б,<,ю>

```

Подробности — в *:help langmap*.

joe

Все проблемы joe можно решить, правильно настроив *locale* (См. раздел "[Настройка локали](#)"), если он правильным образом "пропатчен". Если у вас ничего не получилось, то попробуйте рецепт, описанный ниже. Для того, чтобы распознавать 8-ми битные символы, joe требует специальную опцию *-asis*. Вы можете указать её в командной строке или вставить в файл *~/.joerc* для личного пользования, или в */usr/lib/joerc* для настройки всей системы.

joe, который входит в более-менее современные дистрибутивы правильно распознаёт границы русских слов. А для преобразования регистра *man joe* советует `^K` — фильтровать блок через *tr*, т.е. все сводится к работе *tr*.

Глава 7. Текстовые процессоры

Для публикации документа в печатном виде, как правило, используются специальные программы: текстовые процессоры.

Наиболее мощным для текстовым процессором на текущий момент является *LaTeX* (раздел “[Поддержка кириллицы в TeX/LaTeX](#)”). Это, по сути своей, является скорее решением, нежели программой. Сложности при начальном освоении, вызванные относительно высоким порогом вхождения, компенсируются удивительной гибкостью и лёгкостью решения сложных проблем.

Поддержка кириллицы в TeX/LaTeX

В этом разделе я опишу несколько способов набора кириллических текстов в *TeX* и *LaTeX*. Есть несколько путей для достижения этой цели, которые отличаются в сложности установки и удобстве использования. Например, одна из возможностей — это начать работу без всякой предварительной настройки, используя *Washington AMSTeX Cyrillic fonts*. С другой стороны, вы можете установить пакет *LaTeX*, который легко настраивается на пользование кириллицей.

Вид набранного документа в значительной степени определяется шрифтом, который вы используете.

Основные кириллические шрифты, которые можно использовать при наборе, описаны в разделе “[Шрифты для TeX/LaTeX](#)”.

Обратите внимание, что доступны две версии *LaTeX*, одна из них — *2.09* — старая версия, в то время как *2e* — новая (выпуск *pre-3.0*). Если вы используете *LaTeX 2.09*, то как можно быстрее переходите на *2e*.

Последний сохраняет совместимость со старой версией, но имеет намного больше возможностей.

Последние версии дистрибутивов *LaTeX* (например, “[Кириллизация teTeX](#)” версии старше чем 1.0) включают в себя русификацию *out of the box*. Признаком русифицированности *LaTeX*'а *out-of-the-box* является наличие директории *\$/TEXMF/tex/latex/t2*, *\$/TEXMF* обычно равно */usr/lib/texmf* или */usr/share/texmf*.

Русификация, как нетрудно догадаться, — *T2* (См. раздел “[Пакет T2](#)”).

Пакет T2

Пакет *T2* был изначально создан Werner Lemberg'ом и Владимиром Воловичем. Этот пакет развивается по сей день и, надеюсь, будет развиваться и дальше. В нем присутствуют некоторые недостатки, но в 99% случаев для отображения кириллицы через *LaTeX* его возможностей хватает. Современная русификация *LaTeX* основана именно на этом пакете. Пример русификации через *T2* приведён в разделе “[Кириллизация teTeX](#)”.

Пакет 'T2' можно найти на любом сайте [CTAN](#) в директории *macros/latex/contrib/supported/t2*, последняя версия лежит по адресу ftp.vsu.ru

Официально с 1-го декабря 1998 года дистрибутив *LaTeX2e* поддерживает шрифты со стандартными кириллическими кодировками (T2A, T2B, T2C, X2) и всевозможные входные кодовые таблицы (например, koi8-r, cp1251, cp866, iso8859-5)

BABEL начал поддерживать кодировку *T2* с версии 3.6k (последнюю версию можно взять на любом сайте [CTAN](#) в директории *CTAN:macros/latex/required/babel*).

В файле 'cyrguide.tex', который является частью дистрибутива *LaTeX2e*, описывается настройка и использование кириллицы при работе с *LaTeX*. Для полной поддержки кириллицы требуется иметь *Ih* шрифты и соответствующую таблицу переносов.

Шрифты *Ih* начиная с версии 3.20 (они наверняка есть в вашем дистрибутиве) поддерживают все нужные кодировки —смотрите раздел “[Ih](#)”.

Русские таблицы переносов, поддерживающие кодировку *T2* и оформленные как пакет 'ruhyphen' (он наверняка есть в вашем дистрибутиве), можно найти на любом [CTAN](#) архиве в директории *language/hyphenation/ruhyphen* или по адресу ftp.vsu.ru

В пакете *T2*, кроме непосредственно поддержки *T2* кодировок, дополнительно присутствуют следующие полезные составляющие.

1. Пакет 'mathtext', который позволяет использовать кириллицу в формулах без дополнительных ухищрений. Пакет может быть включён с опцией *warn* — в этом случае он сообщает обо всех случаях использования кириллических букв в формулах. Пакет следует загружать до загрузки пакета *babel* и/или *fontenc*.

2. Пакет `citehack.sty' переопределяет команды LaTeX таким образом, чтобы можно было использовать кириллические символы в качестве аргументов команд \cite и \bibitem — это временный "хак", и в следующей версии LaTeX нужда в нем отпадёт.
3. Скрипт `rubibtex' будет полезен, если вы используете bibtex.
4. Скрипт `rumkidx' будет полезен, если вы используете makeindex — к сожалению, невозможно (без напильника) сделать несколько индексов одновременно.
5. Директория `examples' содержит несколько примеров.
6. Пакет `cyrplain' — на случай, если вы захотите использовать Plain TeX.
7. Директория `cyrfinst' содержит поддержку кириллицы для пакета fontinst.
8. Директория `enc-maps' содержит различные входные кодировки, которые можно использовать при наборе файла LaTeX.

Одной из раздражающих особенностей T2-кириллизации является то, что при "прогонке" tex-файла через latex вывод предупреждений и ошибок ведётся на внутренней кодировке — то есть T2: Артём Чуприн написал программу-фильтр — она перекодирует сообщения об ошибках от "\T2A/" до конца строки, остальное не трогает. Для практических нужд пока хватает. Программу можно найти по адресу <http://www.inp.nsk.su/~baldin/misc/Latex.c>

Так же можно использовать стандартный перекодировщики — например,
 latex myfile.tex | t2filter

Кириллизация teTeX

Если у вас старый дистрибутив teTeXa (версия ниже 1.0), то настоятельно рекомендую скачать последнюю версию этого программного продукта.

Для современных версий TeX'a (teTeX, fpTeX, MikTeX, Web2c TeX) русификация уже входит в дистрибутив. Первое, что нужно сделать — это подключить русские переносы. Для этого нужно отредактировать файл language.dat, находящийся в директории \$TEXMF/tex/generic/config/language.dat (\$TEXMF обычно равно /usr/lib/texmf или /usr/share/texmf), добавив туда строки:

```
ruseng ruenhyph
=rusian
=english
```

Остальные строки нужно закомментировать, либо удалить. После чего необходимо перегенерировать форматные файлы:

```
texconfig init
```

Описанное выше можно проделать через стандартную конфигурационную программу texconfig — выбрать меню hyphenation, далее меню latex и раскомментировать строку со словом russian. При выходе из программы форматные файлы автоматически перегенерируются.

Теперь можно работать с русским языком.

Добавьте в преамбулу документа (после слов \documentclass{})

```
\usepackage{mathtext} % если нужны русские буквы в формулах (не обязательно)
\usepackage[T2A]{fontenc} % внутренняя T2A кодировка TeX
\usepackage[koi8-r]{inputenc} % кодировка - можно использовать [cp866] [cp1251]
\usepackage[russian]{babel} % включение переносов
```

После этого в документе можно использовать русские буквы.

Подробнее о пакете T2 смотрите раздел ["Пакет T2"](#).

P.S. Это только один из вариантов, самый простой и универсальный, решения проблемы кириллизации. Есть и другие варианты и методы, которые на практике могут оказаться более удачными.

Чтобы добавить форматы из *cyrplain* набора в *texconfig*, следует выбрать меню FORMATS и добавить *cyrtxinf* tex language.dat *cyrtxinf.ini* для русского *texinfo*. Аналогично для *cyrblue* и *cyrarbstx*.

Чтобы добавить формат *cyrtex* (русский TeX), проще в каталоге *web2c* выполнить:
`initex "input cyrtex.ini 'dump'`

и бросить символьическую ссылку с именем *cyrtex* на *tex* (например, в */usr/bin*).

Чтобы переключится на русский в этих plain форматах, надо использовать команду *\language N* где N — номер, под которым числится русский язык в получившейся раскладке.

Стиль

В этом разделе даны рецепты, как создать и/или модифицировать свой стиль, естественно, это относится к использованию кириллицы в *LaTeX*. Предложенные рецепты в основном опираются на пакет русификации *T2* (См. раздел [“Пакет T2”](#)).

Кириллические буквы в формулах

Для того, чтобы использовать кириллицу в формулах следует в преамбуле до задания *inputenc*, *fontenc*, *babel* добавить строчку:
`\usepackage{mathtext}`

Что бы сделать кириллические буквы в формулах наклонными, то нужно в преамбуле добавить строчку:
`\DeclareSymbolFont{T2Aletters}{T2A}{cmr}{m}{it}`

Подписи в картинках

Для создания кириллических подписей в картинках следует воспользоваться пакетом *psfrag*. Подробнее про этот пакет описан в разделе [“Пакет psfrag \(eps\)”](#).

Texinfo

Чтобы кириллицу можно было использовать с *texinfo* следует воспользоваться пакетом *cyrplain* (См. раздел [“Кириллизация teTeX”](#)).

Кириллица и bibtex

Что бы можно было использовать кириллицу с *bibtex*'ом — то есть иметь возможность написать что-то вроде:

`\bibitem{Иванов1000}`

Следует воспользоваться пакетом *citehack* — добавить в преамбуле:
`\usepackage{citehack}`

Замечание: По умолчанию кириллицу нельзя использовать в качестве ИНДЕКСА (т.е. внутреннего объекта, который никак не отображается в свёрстанном документе). В необязательных аргументах *\cite* и *\bibitem* (которые отображаются в документе) использовать кириллицу можно. Пакет *citehack* позволяет использовать 8-битные символы в качестве индекса.

Кириллица в списках

Чтобы заставить окружение *enumerate* нумеровать списки *\item* по-русски. Следует воспользоваться пакетом *babel*:

- Маленькие буквы:

`\renewcommand{\theenumi}{\asbuk{enumi}}`

- Большие буквы:

`\renewcommand{\theenumi}{\Asbuk{enumi}}`

Эти команды делают нужную нам нумерацию на верхнем уровне вложенности. На следующем уровне идут *enitmii*, *enitmiii*, *enitmiv*.

Кириллица и нумерация приложений

Для нумерации приложений русскими буквами (некоторые как раз любят латинские) следует написать:

```
\makeatletter
\renewcommand\appendix{\par
\setcounter{section}{0}%
\setcounter{subsection}{0}%
\gdef\thesection{\@Asbuk\c@section}%
\makeatother
```

Перенос формул

Чтобы символы бинарных операций ("+", "-", "=") в *LaTeX*, набранные в строчных формулах, при переносе на другую строку, дублировались например:

```
a + b =  
= c
```

МОЖНО воспользоваться такими командами в преамбуле:

```
%begin{latexonly}
\renewcommand\neg{\mathchar"3236\mathchar"303D\nobreak
  \discretionary{} {\usefont
    {OMS}{cmsy}{m}{n}\char"36\usefont
    {OT1}{cmr}{m}{n}\char"3D} {}}
\begin{group}
\catcode`+`active\gdef+{\mathchar8235\nobreak\discretionary{}%
  {\usefont{OT1}{cmr}{m}{n}\char43} {}}
\catcode`-`active\gdef-{\mathchar8704\nobreak\discretionary{}%
  {\usefont{OMS}{cmsy}{m}{n}\char01} {}}
\catcode`=`active\gdef={\mathchar12349\nobreak\discretionary{}%
  {\usefont{OT1}{cmr}{m}{n}\char61} {}}
\end{group}
\def\cdot{\mathchar8705\nobreak\discretionary{}%
  {\usefont{OMS}{cmsy}{m}{n}\char1} {}}
\def\times{\mathchar8706\nobreak\discretionary{}%
  {\usefont{OMS}{cmsy}{m}{n}\char2} {}}
\mathcode`==32768
\mathcode`+=32768
\mathcode`-=32768
%end{latexonly}
```

Первая и последняя строка комментария нужна в случае, если вы используете *latex2html* для публикации вашего файла в html формате.

Дополнение: Вероятно, эти команды будут включены в планируемый пакет-дополнение к *babel* (добавляющий некоторые особенности русскоязычного набора). Кто заметит глюки в вышеупомянутых командах — дайте знать Владимиру Воловичу vvv@vvv.vsu.ru или 2:5025/2.72.

Перенос

Чтобы правильно писать составные слова, типа, "коммерческо-посредническая". следует добавить следующие строчки (См. README от пакета *ruhyphen*):

```
\lccode`-=`-
\defaulthyphenchar=127
% ...
\usepackage[T2A]{fontenc}
```

Если нужно явно задать правило переноса для какого-то составного слова, то вставьте что-то вроде:

```
\begin{group}
\lccode`-=`-
% Пример: задать правила переноса для `коммерческо-посредническая':
\hyphenation{ком-мер-че-ско-по-сред-ни-че-ская}
\end{group}
```

Отступ для первого параграфа раздела

В русской литературе, в отличии от английской, принято, чтобы первый параграф раздела имел точно такой же отступ, как и везде. Для этого следует воспользоваться пакетом *indentfirst*.

```
\usepackage{indentfirst}
```

Работа с Washington Cyrillic

Этот пакет был создан для Американского Математического Общества, чтобы дать возможность создавать документы со ссылками на первоисточники на русском. Следовательно, авторы не очень "напрягались" при

создании этого пакета, и шрифты в результате этого выглядят довольно неуклюже. Обычно этот пакет упоминается как "по настоящему плохой пакет кириллицы для TeX". Однако мы обсудим его, так как он очень прост в использовании и не требует установки — этот набор содержится в большинстве дистрибутивов TeX. Конечно, у вас не будете такой роскоши, как автоматическая расстановка дефисов, но всё равно...

- Снабдите ваш документ следующими директивами:
 - `\input cyracc.def`
 - `\font\tencyr=wncyr10`
 - `\def\cyr{\tencyr\cyracc}`
- Теперь, чтобы напечатать символы кириллицы, вставьте `\cyr` для печати используйте соответствующий латинский символ или команду TeX. То есть строчные буквы русского алфавита соответствуют следующим сочетаниям:
 - `a b v g d e \e zh z i \u i k l m n o p r s t u f kh c ch sh shch`
 - `\cprime} y \cdprime} \e uy ya`

Чрезвычайно сложно преобразовывать ваши русские тексты в такую кодировку, но вы можете автоматизировать этот процесс. Программа `translit` (раздел "[Символьная перекодировка](#)") поддерживает опцию вывода TeX.

OpenOffice

Возьмите уже русифицированную версию OpenOffice, например с [ftp AltLinux](#). Для корректной печати и совместимости с MS Word следует установить ttf шрифты (раздел "[Шрифты TrueType](#)") с помощью скрипта `spadmin`.

StarOffice

Вместо StarOffice следует воспользоваться программным пакетом OpenOffice (раздел "[OpenOffice](#)").

nroff

Для того, чтобы через nroff можно было "пропустить" символы кириллицы, надо использовать его с ключом `-Tlatin1`. Пропишите где-нибудь в стартовом скрипте (если у вас bash, то в `.bashrc`)
`alias nroff='nroff -Tlatin1'`
Просмотр файлов в mc запускается через nroff (по крайней мере, в mc, идущем со Slackware'96). В файле `/usr/lib/mc/mc.ext` следует в строке вызова nroff'a изменить параметр вызова с `-Tascii` на `-Tlatin1`.

Глава 8. Вывод на печать и PostScript

Напечатать что-либо — это всегда проблема. Имеется набор различных принтеров от различных производителей с различными особенностями. Даже для вывода на печать обычного ASCII текста нет никаких общих решений (это применимо не только к UNIX, но также и к другим операционным системам). Принтеры имеют различные управляющие языки, и очень часто они имеют сильно различные подходы к поддержке иностранных языков. Хорошая новость — это то, что сейчас в качестве управляющего языка, как стандарт de facto для описания работы печати, используется язык *PostScript*, разработанный [Adobe Corporation](#). Много принтеров имеют встроенный *PostScript* интерпретатор, то есть вам надо просто послать на принтер Postscript данные. Для тех, у кого этого нет, существует *программные PostScript интерпретаторы*, которые берут данные PostScript и преобразовывают в специфический для данного принтера управляющий код. Один из них мы сейчас обсудим (вероятно, лучший из лучших). Это GNU GhostScript (gs для краткости).

Другая проблема — это широкий спектр требований, предъявляемых к печати. Например, иногда вы хотите просто напечатать часть вашей программы на "C", содержащей в качестве комментария текст на русском, так что вы не нуждаетесь в "навороченной" процедуре печати — вам нужен простой ASCII вывод с одним шрифтом. Совсем другое дело, когда вы создаёте открытку для вашей подруги. В этом случае вы, вероятно, будете нуждаться в печати документа с различными шрифтами и т.д. И это уже определённо требует больших усилий по установке поддержки кириллицы.

Чтобы выполнить вышеупомянутую задачу по выводу С программы, вы должны заставить ваш принтер понять только *один* шрифт кириллицы и (возможно) установить некоторую программу-фильтр, чтобы выводить данные в соответствующем формате. Чтобы совладать со второй задачей, вы должны обучить ваш принтер различным шрифтам и иметь специальное программное обеспечение.

Бывают задачи, требующие для выполнения нечто среднее, тогда вам нужна программа, которая знает как организовать и шрифты, и соответствующий вывод в принтер, так что вы можете, скажем, получить на выходе качественно отпечатанный текст без сложных систем подготовки текстов.

Преобразование текста в PostScript

Иногда у вас есть простой ASCII KOI8 текст, и вы хотите его только напечатать. Один из самых простых способов это сделать — это воспользоваться услугами программ, преобразующих текст в PostScript.

Есть ряд программ, делающих такое преобразование. Я лично предпочитаю [a2ps](#). Первоначально разработанная как простой преобразователь text-to-PostScript, сейчас эта программа стала "матерью", легко настраиваемым программным продуктом с большим количеством опций и настроек. Она позволяет управлять форматами, размещением страниц, выделением и т.д. Другая утилита, делающая примерно то же самое (теперь доступная как часть проекта *GNU*) — [enscript](#).

A2ps конвертер

Преобразователь текста в PostScript был и остаётся одним из наиболее универсальных средств печати. Все, в чем вы теперь нуждаетесь — это PostScript принтер.

Команда, которую я использую для этого:

```
a2ps -X koi8r --print-anyway <файл>
```

Комплект шрифтов i18n-fonts нужных для преобразования кириллических текстов в Postscript не входит в дистрибутив a2ps начиная с версии 4.12 *ПО ПРИЧИНЕ НЕ ОТКРЫТОЙ ЛИЦЕНЗИИ*. Шрифты следует скачать с домашней странички [a2ps](#) или с [ftp a2ps](#).

Для работы a2ps можно использовать шрифты [sharatype-2.0.tar.gz](#), где в комплекте идёт все необходимое для прикрутки оных шрифтов к a2ps. Будет работать и в *Latin-1* и в *KOI8*. Должно также и в *Latin-2*.

Следует учитывать, что программы имеет весьма развитый "интеллект" и определяет своё поведение в зависимости от структуры файлы, поэтому лучше всего привести текст перед печатью в "плоский" вид, то есть убрать все внутренние теги, которые могут смутить программу.

GNU enscript

Программа GNU enscript, как и a2ps, была разработана для преобразования текста в PostScript, и она также поддерживает не-ascii кодировки. Программа не имеет русских PostScript шрифтов в своём составе, но их очень просто доустановить. Как это сделать, описано ниже:

1. Установите последний enscript. Самую последнюю версию 1.5. вы можете найти в [GNU FTP архиве](#).
2. Если вы — пользователь RedHat Linux, загрузите и установите [шрифт Cyrillic Textbook](#).
3. Если вы не используете RPMки, вытяните файл textbook.tar.gz из архива на [sunsite.unc.edu](#) (здесь лежит программное обеспечение для русификации). Разархивируйте этот файл в каталог, где размещены шрифты для enscript (обычно /usr/share/enscript). Теперь перейдите туда и выполните следующую команду:
 4. mkafmmap *.afm
5. Установка завершена. Попробуйте напечатать текст в KOI8-R следующей командой:
 6. enscript --font=Textbook8 --encoding=koi8 some.file

rtxt2ps

Если же вам требуется действительно быстрый и простой способ, а качество вывода для вас не критично, и все, что вам нужно — это только русский текст на бумаге, попробуйте пакет [txt2ps](#). Это очень простой, без украшательств, конвертер текста в PostScript. Качество вывода не очень хорошее (честно говоря, *плохое*), но это работает.

Преобразование текста в TeX

Если все, в чем вы нуждаетесь — это печать ASCII текста без дополнительной обработки, то вы можете использовать некоторые программки, которые могут преобразовать ваш текст кириллицы в готовый TeX файл. Одна из самых лучших программ для таких целей — это translit (Смотрите раздел “[Символьная перекодировка](#)”). В этом случае вы даже не должны беспокоиться относительно установки шрифтов кириллицы для TeX, так как translit использует пакет кириллицы Washington Cyrillic, который включен в большинство дистрибутивов TeX (подробнее об этом пакете смотрите в разделе “[Работа с Washington Cyrillic](#)”).

Кириллица в PostScript

Чтобы печатать русский текст с использованием PostScript, вы должны удостовериться относительно следующих вещей:

- шрифт кириллицы *загружен* в принтер, и тогда у вас всё уже должно работать.
- текст кириллицы включен в документ — здесь можно воспользоваться следующим способом: Для того, чтобы вшить русские шрифты в сам ps-файл, нужно после printtool в соответствующем файле /var/spool/lpd/lp.../postscript.cfg заменить строку 'GSDEVICE=POSTSCRIPT' на 'GSDEVICE=pswrite'. Кроме описанного метода, можно включить шрифты в файл PostScript руками с помощью обычного текстового редактора — подробно это описано в разделе “[Инкапсуляция шрифтов Type1 в Encapsulated PostScript \(eps\)](#)”.
- текст кириллицы использует символьные коды, которые соответствуют требованиям шрифта.
- чтобы печатать текст кириллицы, *выбран* соответствующий шрифт.

Не имеется никакого достаточно общего решения, чтобы рекомендовать его как окончательное. Я попробую осветить различные способы для решения различных проблем, связанных с этим вопросом. Один способ — это побороть проблемы установки кириллицы вообще, и состоит он в том, чтобы использовать [Ghostscript](#). Ghostscript (или просто gs от newspeak) абсолютно free (ну, не совсем) интерпретатор PostScript. У него есть много преимуществ; среди них:

- Способность работать на многих платформах (различные Unix, MS Windows и т.д);
- Поддержка для огромного количества не-PostScript принтеров;
- Высокая степень настраиваемости.

В нашем специфическом случае является важным то, что однажды установив и настроив Ghostscript, мы можем все печатать через него, таким образом нам не надо дополнительно настраивать другие PostScript устройства (например, *HP LaserJet IV*).

Инкапсуляция шрифтов Type1 в Encapsulated PostScript (eps)

Рисунки в формате PostScript часто требуется включать в текст, подготовленный таким средством форматирования, как TeX. Компания Adobe разработала формат файла Инкапсулированный PostScript (EPS или EPSF), который создаётся в соответствии с "Соглашениями по структурированию документов в формате PostScript" (PostScript Document Structuring Conventions).

Большинство современных графических приложений порождает файл в общепринятом формате EPS, который без труда может быть использован LaTeХом. Существует одно но: как правило, при создании EPS файла шрифты, которыми делаются надписи (шрифты, конечно же, — PostScript Type1), в него не включаются и при печати заменяются первыми попавшимися.

Если у вас есть картинка в EPS, использующая шрифт AAA, и файл AAA.PFA с этим PostScript шрифтом, то для того, чтобы инкапсулировать шрифт в картинку, надо с помощью любого текстового редактора вставить файл AAA.PFA целиком непосредственно перед строчкой %%EndProlog в файл EPS.

Если у вас есть только PFB шрифты, то их следует его преобразовать в PFA. Для этого воспользуйтесь утилитой pfb2pfa (см. [CTAN](#)) или t1ascii (см. [t1utils](#)).

Для вставки вместо редактора можно использовать скрипт includeresources из psutils — он делает то же, но автоматически^[6].

Этот же приём срабатывает и для обычного PostScript файла.

Пакет psfrag (eps)

Если вы хотите использовать картинки, сделанные, например, в xfig при работе с TeX/LaTeX, то Вам поможет пакет psfrag. В xfig пишете вместо надписей тэги — типа Petr, moment и т.д. Экспортируете картинку в eps. А в TeХовском файле пишете

```
\usepackage{psfrag}  
\psfrag{Petr}{Здесь будет город заложен\!dots}  
\psfrag{moment}{Я помню чудное мгновенье}
```

и т.д. Тогда в результате будут надписи в том же шрифте, что основной текст.

Преимущества такого подхода:

- надписи в том же шрифте, что и текст;
- можно вставлять любые формулы и т.д. — все, что выдержит TeX.

Недостатки:

- правильное расположение текста получается методом проб и ошибок;
- если надо результат не в PostScript, а в pdf, то приходится потрудиться: создавать односторонние документы в TeХе (с псфраговскими фрагментами), их переводить в eps, оттуда в pdf — в общем, возиться, но так как Вы в Linux, то написать скрипт, который будет это делать за вас, я думаю, труда не составит.

Ghostscript

Пакет Ghostscript является совершенно уникальным продуктом — фактически все, что печатается в Unix, проходит через Ghostscript.

Настройка ghostscript

Иногда необходимо иметь какую-нибудь специфическую, не общесистемную настройку ghostscript. Например, возможно, что у вас нет привилегий root-а, или вы не желаете воспользоваться ими из соображений безопасности. В системе могут совместно работать пользователи, нуждающиеся в разных настройках. Кроме того, универсальной настройки просто не существует. Например, для того, чтобы Netscape печатал кириллицу, нужно заменить стандартные латинские PostScript шрифты кириллическими. Но с другой стороны, после этой замены будет невозможно напечатать (и даже увидеть на экране) некоторые латинские тексты.

Создайте новый каталог (например, `~/ghostscript`) и поместите в него все новые кириллические шрифты. Скопируйте в него также файл Fontmap и затем измените его как Вам угодно. Установите переменную окружения `GS_LIB`:

```
export GS_LIB=~/ghostscript
```

Эта переменная указывает ghostscript, что нужно сначала заглянуть в каталог `~/ghostscript` (где находятся ваши новые шрифты и файл Fontmap), и лишь потом в общесистемные каталоги.

Добавление шрифтов кириллицы к Ghostscript

Это важно, так как вы, вероятно, не захотите взваливать ответственность за включение шрифтов кириллицы в PostScript на другие программы. Взамен вы добавляете их только к gs и заставляете программы выводить русский текст, совместимый с этими шрифтами.

Чтобы добавлять новый шрифт (в формате pfa или pfb) в gs, вы должны:

1. Поместить этот шрифт в каталог шрифтов tt/gs/. Обычно это `/usr/lib/ghostscript/fonts` — можно уточнить с помощью команды `'gs -h'`.
2. Добавить соответствующие имена и aliases для шрифта в файле Fontmap в каталоге gs.

О том, где можно найти шрифты Type1, написано в разделе “[Шрифты Type1](#)”.

Использование старого матричного принтера для печати кириллического текста

Если у вас есть старый добрый матричный принтер, и вы нуждаетесь в простом выводе текста на KOI8, то попробуйте следующее:

1. Найти соответствующий KOI8 шрифт для вашего принтера. Проверьте ftp архивы MS DOS в Internete (например, посмотрите на [архив SimTel](#)).
2. Прочитайте руководство и найдите в нем описание загрузки такого шрифта в ваш принтер. Напишите простенькую программу, делающую это.
3. Запускайте эту программку из соответствующего rc файла при загрузке.

Таким образом, наличие символов Cyrillic в верхней части набора символов принтера позволит вам печатать тексты по-русски без дополнительных ухищрений.

Альтернативно к KOI8-R шрифтам вы можете попробовать использовать CP866 шрифты. Для этого имеются две причины:

- Вероятно, найти CP866 шрифты намного проще, так как те были очень широко распространены во времена MS-ДОС.
- Наличие соответствующего CP866 шрифта позволит вам печатать также и псевдографические символы.

Следует отметить, что в Болгарии вместо CP866 используется кодировка МИК.

Однако, в этом случае вы должны будете преобразовать ваши тексты из KOI8-R в CP866 перед посылкой их на принтер. Это не проблема, так как имеется множество программ, делающих это (для примера смотрите раздел “[Символьная перекодировка](#)”), так что для этого вам нужно только вызвать такую программку из файла `/etc/printcap` в “if поле”. Например, с программой translit можно сделать следующее:

```
if=/usr/bin/translit -t koi8-alt.rus
```

Для подробностей смотрите printcap(5).

[6] Мне так и не удалось разобраться с работой этого скрипта, поэтому буду благодарен по поводу любой информации.

Глава 9. Кириллизация программ

В этом разделе описывается кириллизация популярных программ. К сожалению пока не все программы, даже популярные, соответствуют стандарту локализации и интернационализации (раздел [Глава 5. Локализация и Интернационализация](#)).

Кириллизация shell

bash

Для того, чтобы заставить bash понять 8-ми битные символы, необходимо установить три переменные. Лучше всего это сделать из файла `~/.inputrc`.

Должны быть сделаны следующие установки:

```
set meta-flag on
set convert-meta off
set output-meta on
```

Следует отметить, что файл `~/.inputrc` не имеет отношения непосредственно к `bash` — это конфигурационный файл библиотеки *GNU readline* (раздел [“GNU readline”](#)), используемой `bash`-ем. Поэтому совет о `.inputrc` кириллизует не только `bash`, но и многие другие программы (все, использующие *GNU readline*).

csh/tcsh

Поместите в `.cshrc` следующее:

```
setenv LC_CTYPE iso_8859_5
stty pass8
```

Если у вас нет POSIX-совместимого `stty` (только не в Linux), замените последнюю строчку следующей:
`stty -istrip cs8`

zsh

Сделайте то же самое, что делается для `csh` ([“csh/tcsh”](#)). Файл инициализации в этом случае — `.zshrc` или `/etc/zshrc`.

ksh

Что касается public domain реализации `ksh` — `pdksh` версии 5.1.3 и выше, то вы можете разрешить 8-ми разрядный ввод только для `vi` в режиме ввода. Для этого используйте:
`set -o vi`

Поддержка кириллицы в perl

В этом разделе описывается методы позволяющие работать с кириллицей из скриптов, написанных языке `perl`

Поддержка locale в perl

Для того чтобы можно было выполнять регексы на кириллических символах с ключом "i" (а так же использовать стандартные функции преобразование текста — такие как `uc`, `lc`, `sort` и т.д.) в системах с корректной локализацией (FreeBSD, Linux на glibc 2.1) это делается так:

```
use locale;
use POSIX qw(locale_h);
setlocale(LC_CTYPE, 'ru_RU.KOI8-R');
```

(можно ещё выставить переменную окружения export LANG=ru_RU.KOI8-R тогда будет достаточно вызова use locale; в Perl-скрипте)

Символьная перекодировка средствами perl

Ниже идут примеры

- Пример 1:

```
• sub wintokoi {  
•     my $pvdcoderwin=shift;  
•     $pvdcoderwin=~  
•         tr/\xC0\xC1\xC2\xC3\xC4\xC5\xC6\xC7\xC8\xC9\xCA\xCB\xCC\xCD\xCE\xCF\xD0\xD1\xD2  
•         \xD3\xD4\xD5\xD6\xD7\xD8\xD9\xDA\xDB\xDC\xDD\xDE\xDF\xE0\xE1\xE2\xE3\xE4\xE5  
•         \xE6\xE7\xE8\xE9\xEA\xEB\xEC\xED\xEE\xEF\xF0\xF1\xF2\xF3\xF4\xF5\xF6\xF7\xF8  
•         \xF9\xFA\xFB\xFC\xFD\xFE\xFF\xE1\xE2\xF7\xE7\xE4\xE5\xF6\xFA\xE9\xEA\xEB\xEC  
•         \xED\xEE\xEF\xF0\xF2\xF3\xF4\xF5\xE6\xE8\xE3\xFE\xFB\xFD\xFF\xF9\xF8\xFC\xE0\xF1  
•         \xC1\xC2\xD7\xC7\xC4\xC5\xD6\xDA\xC9\xCA\xCB\xCC\xCD\xCE\xCF\xD0\xD2\xD3\xD4\xD5  
•         \xC6\xC8\xC3\xDE\xDB\xDD\xDF\xD9\xD8\xDC\xC0\xD1;  
•     return $pvdcoderwin;  
• }
```

- Пример 2:

```
• sub dostoiso {  
•     my $pvdcoderwin=shift;  
•     $pvdcoderwin=~  
•         tr/\x80\x81\x82\x83\x84\x85\x86\x87\x88\x89\x8A\x8B\x8C\x8D  
•         \x8E\x8F\x90\x91\x92\x93\x94\x95\x96\x97\x98\x99\x9A\x9B\x9C\x9D\x9E\x9F\xA0  
•         \xA1\xA2\xA3\xA4\xA5\xA6\xA7\xA8\xA9\xAA\xAB\xAC\xAD\xAE\xAF\xE0\xE1\xE2\xE3\xE4  
•         \xE5\xE6\xE7\xE8\xE9\xEA\xEB\xEC\xED\xEE\xEF\xB0\xB1\xB2\xB3\xB4\xB5\xB6\xB7  
•         \xB8\xB9\xBA\xBB\xBC\xBD\xBE\xBF\xC0\xC1\xC2\xC3\xC4\xC5\xC6\xC7\xC8\xC9\xCA\xCB  
•         \xCC\xCD\xCE\xCF\xD0\xD1\xD2\xD3\xD4\xD5\xD6\xD7\xD8\xD9\xDA\xDB\xDC\xDD\xDE\xDF  
•         \xE0\xE1\xE2\xE3\xE4\xE5\xE6\xE7\xE8\xE9\xEA\xEB\xEC\xED\xEE\xEF/;  
•     return $pvdcoderwin;  
• }
```

Остальное получается по аналогии.

Примером, перекодировщика на perl с автоматическим определением исходной кодировки может служить [программа 2k](#).

Символьная перекодировка

В Internet можно найти множество программ, перекодирующих кириллические тексты. Каждая из них имеет свою изюминку и отличается степенью поддержки кириллицы.

Утилиты перекодировки должны быть стандартными. В нашем, частном, случае такой "стандартной" утилитой является iconv (входит в поставку glibc) или [GNU recode](#).

Например, чтобы перекодировать из cp1251 в koi8-r, следует сказать:

```
$ iconv -fCP1251 -tKOI8-R -o<outfile> infile
```

или

```
$ recode CP1251..KOI8-R winfile.txt
```

Внимание!: Для перекодировки в/из KOI8-U в GNU recode надо использовать *koi8-ru*, под *koi8-u* программа понимает что-то другое.

Если же, собственно, перекодировке требуется автоматически определить исходную кодовую страницу, то можно воспользоваться [утилитой 2k](#).

В конце концов сгодится обычный lynx (links) (cp1251 в koi8-r):

```
$ lynx -assume_local_charset cp1251 file.txt
```

less

Если локаль (см. раздел “[Настройка локали](#)”) не настроена^[2], то вывод кириллицы через less можно получить, установив переменную окружения LESSCHARSET:

```
export  
LESSCHARSET=koi8-r
```

Это решение годится для *всех* 8-битовых кириллических кодировок.

Старые версии less не поддерживали эту установку, поэтому вместо koi8-r в этом случае следует подставить latin1. При этом однако становятся недоступными символы, чьи коды находятся в диапазоне 128-159. К счастью, коды букв обычно вне этого диапазона.

При *настроенной локали* (см. раздел “[Настройка локали](#)”) указывать LESSCHARSET *НЕ НАДО*. Более того, в `~/.lesskey` надо добавить

```
#env  
LESSCHARSET=
```

чтобы программа игнорировала установку LESSCHARSET= другими “глупыми” программами (к примеру, man). После этого надо запустить lesskey для получения бинарного файла `~/.less`. В противном случае он не будет вызывать `setlocale(LC_CTYPE,"")` и, как следствие, не будет i-case search для русских букв.

В качестве полумеры можно воспользоваться ключом `g` или `row-control-chars`. При использовании этого ключа возможен просмотр не ASCII символов в любых условиях, но при этом возникают побочные эффекты, описанные в man less.

mc (The Midnight Commander)

Чтобы увидеть кириллицу из *mc*, выберите опцию full 8 bits в Options/Display меню.

Если и в этом случае при просмотре файла из *mc* вы не увидите кириллических символов, то загляните в раздел “[nroff](#)”.

В случае возникновения проблем при просмотре man статей — сверьтесь с разделом “[man](#)”.

Если у вас проблемы в виде уродливых оконных рамок, то проконсультируйтесь в разделе [Глава 3. Консоль](#).

man

Сейчас довольно бурно развивается деятельность по переводу всего и вся на русский язык (ну, не так чтобы уж очень, но кое-кто кое-что делает). Все больше и больше появляется статей *man*, переведённых на русский язык, но вот отобразить их не всегда удаётся.

Если есть файл с русскими буквами, то вызов, скажем *man ar* приводит к ерунде на экране. Для исправления этого безобразия следует поправить соответствующие строки в `/usr/lib/man.config`, если это файл есть, или правильно настроить *less* (см. раздел “[less](#)”).

ls

Если локаль установлена неправильно (см. раздел “[Настройка локали](#)”), то ls не будет печатать кириллические символы. В этом случае, возможно, поможет одна из следующих команд: `ls -N`, `dir -N` или `ls --show-control-chars`.

rlogin

Удостоверьтесь, что shell на месте адресата установлена правильно. Если ваш rlogin не работает как надо по умолчанию, то используйте ‘`rlogin -8`’.

Пропишите в стартовом скрипте (если вы используете bash, то это `.bashrc`)

```
alias rlogin=rlogin -8'
```

telnet

Если возникают проблемы с вводом русских символов, надо создать файл `~/.telnetrc` со следующей строкой:

```
DEFAULT set outbinary
```

Вы можете встретить проблемы при работе в кодировке cp1251 — не передаётся маленькая русская буква “я” 0xff. У протокола TELNET 0xff — это первый символ управляющей последовательности. Дабы передать собственно “я”, нужно его удваивать: 0xff, 0xff. При использовании KOI8-R такая проблема отсутствует.

off-topic: Вместо telnet следует использовать ssh (Secure Shell). Протокол telnet позволяет получать информацию о переданной с помощью него информации просто прослушивая сеть, следовательно, является опасным в смысле безопасности.

IrcII

Добавить в файл конфигурации `~/.ircrc` следующие строчки:

```
/set translation russian  
/set eight_bit_characters on
```

tgif

Возможно, наиболее правильным способом кириллизации подписей в рисунках, является использование пакета “[Пакет psfrag \(eps\)](#)”. Если вы по какой-то причине не используете этот пакет, то можно воспользоваться описанным ниже рецептом.

В `~/.Xdefaults` прописываются следующие строчки:

```
Tgif.AdditionalFonts: \n\  
Times koi8-medium-r-normal \n\  
koi8-r \n\  
TimesKOI8 \n\  
\n\  
Times koi8-bold-r-normal \n\  
koi8-r \n\  
TimesKOI8-Bold \n\  
\n\  
Times koi8-medium-i-normal \n\  
koi8-r \n\  
TimesKOI8-Italic \n\  
\n\  
Times koi8-bold-i-normal \n\  
koi8-r \n\  
TimesKOI8-BoldItalic \n\  
\n\  
аналогично-для-Arial \n\  
аналогично-для-Courier
```

- первая строка — экранный шрифт,
- вторая — его кодировка,
- третья — ps-font.

У меня и там, и там используются одни и те же ps-фоントы, что обеспечивает полный визивиг.

P.S. tgif 4.1.22 — Почему-то понимание русского появляется не после `-D_LOCALE_SUPPORT`, а после `-D_ENABLE_NLS`, который, по идеи, включает использование локализованных .po файлов. Ну да ладно — главное что работает.

xfig

Возможно, наиболее правильным способом кириллизации подписей в рисунках, является использование пакета “[Пакет psfrag \(eps\)](#)”. Если вы по какой-то причине не используете этот пакет, то можно воспользоваться описанным ниже рецептом.

Следует установить последнюю версию программ fig2dev и xfig и запустить программу с ключом `-international`, или прописать в `/usr/X11R6/lib/X11/ru/app-defaults/Fig`:

```
Fig*international: false
```

```
Fig.international: true
```

```
Fig.inputStyle: Root
```

Чтобы работала печать, нужно скачать файл [ru_RU.KOI8.ps](#). Файл нужно поместить в директорию `/usr/X11R6/lib/X11/fig2dev/`. Для удобства файл нужно переименовать в `russian.ps` и создать два символьических линка.

```
mv ru_RU.KOI8.ps /usr/X11R6/lib/X11/fig2dev/russian.ps  
cd /usr/X11R6/lib/X11/fig2dev/  
ln -s russian.ps ru_RU.KOI8.ps  
ln -s russian.ps ru_RU.KOI8r.ps
```

В результате проведённых манипуляций становится возможной печать кириллицы прямо в окне xfig'a, полностью исключается промежуточная стадия в TeX'.

Недостатки: ввод кириллицы кривой, а именно: нельзя вводить символы в окне edit, для настройки шрифтов требуются проделать дополнительные действия с файлом конфигурации.

Для того чтобы этот рецепт сработал, нужно хитро скомпилировать пакеты xfig и fig2dev, но даже в дистрибутиве RedHat 7.2 они такие и есть, а уж в "русских" дистрибутивах — и подавно.

Подробнее об интернационализации пакета можно прочитать здесь:
<http://member.nifty.ne.jp/tsato/xfig/latin/latin-e.html>.

cygwin

К особенностям кириллизации cygwin относится то, что в нём отсутствует локаль, поэтому для возможности работы с символами кириллицы следует настраивать каждое приложение отдельно.

Для начала следует настроить библиотеку readline (раздел "[GNU readline](#)".

Для просмотра текстов следует настроить less (раздел "[less](#)"

[?] Локаль должна быть настроена. Как это сделать подробно расписано в разделе "[Locale](#)"

Глава 10. Совместимость

Следовать стандарту — это ещё не все. В реальной жизни надо обеспечить ещё и обратную совместимость. В нашем случае это означает, что наши настройки не должны препятствовать созданию данных с помощью других кодировок, отличных от стандартной. Это могут быть данные в *CP866* или в *CP1251*. Также должна быть возможность запускать русскоязычные программы из MS-DOS.

В большинстве случаев (кроме HTTP) достаточно обеспечить преобразование данных в *KOI8-R*. Если мы говорим о данных с простой структурной организацией, то это просто —смотрите раздел "[Символьная перекодировка](#)".

Другое дело — данные с определённой структурой. В этом случае действия не так тривиальны. Я попробую описать стандартные подходы для решения этой проблемы.

Совместимость данных в MIME формате

MIME формат — стандартное архитектурно-независимое представление данных. Первоначально это представление данных было разработано для письменных сообщений, а сейчас его используют и в других местах. Стандарт MIME определяет формат, который открыт для расширений и позволяет поддержку и работу со специфическими данными. Например, если я послал письмо, содержащее *MIME объект* video/mpeg типа (MPEG файлы), моя программа приёма электронных сообщений автоматически декодирует его и запустит MPEG проигрыватель. Описание MIME можно найти на [web странице IANA](#).

Большинство UNIX программ, предлагающих MIME сервис, для этих целей используют пакет metamail, который содержит набор утилит и файлов данных для работы с объектами MIME. Несколько файлов конфигурации (*/etc/mailcap* для системной настройки и *~/.mailcap* — для пользовательской настройки) определяют директивы для работы с объектами MIME различных типов.

Поэтому если вы получили поток MIME данных, содержащий текст в одной из устаревших кодировок, вы можете определить соответствующие MIME-директивы для конвертации такого текста в *KOI8*.

Ниже перечислены MIME-законы, которые описывают правила работы с обычными текстами и текстами в richtext формате, использующими другие кодировки, отличные от *KOI8*. Вы можете вставить эти директивы в один из файлов конфигураций MIME.

Обратите внимание: Эти директивы используют пакет translit для того, чтобы производить само преобразование. Для более полной информации об этой программе и для информации по перекодировкесмотрите раздел "[Символьная перекодировка](#)".

```
text/plain; translit -t cp1251-koi8.rus <%s; test=test \
    "echo %{charset} | tr '[A-Z]' '[a-z]'" = cp1251; copiousoutput

text/richtext; translit -t cp1251-koi8.rus <%s; test=test \
    "echo %{charset} | tr '[A-Z]' '[a-z]'" = cp1251; copiousoutput

text/plain; translit -t alt-koi8.rus <%s; test=test \
    "echo %{charset} | tr '[A-Z]' '[a-z]'" = cp866; copiousoutput

text/richtext; translit -t alt-koi8.rus <%s; test=test \
    "echo %{charset} | tr '[A-Z]' '[a-z]'" = cp866; copiousoutput

text/plain; translit -t alt-koi8.rus <%s; test=test \
    "echo %{charset} | tr '[A-Z]' '[a-z]'" = alt; copiousoutput

text/richtext; translit -t alt-koi8.rus <%s; test=test \
```

```
``echo %{charset} | tr '[A-Z]' '[a-z]'" = alt; copiousoutput
```

Обычно это работает только в случае текста. Бинарные файлы данных должны сами отрабатывать данные в различных кодировках (по крайней мере, это должны делать программы, создавшие их). Поэтому если вы послали файл Microsoft Word в кодировке *CP1251*, то с этим должна разобраться программа, в которой вы читаете этот текст (Например MS Word или Applix Words).

К несчастью, действительная ситуация далека от идеала. Много программных продуктов имеют собственные идеи по поводу того, как использовать MIME.

Кириллические имена файлов в файловой системе MS Windows

Windows имеет возможность давать имена файлов в кодировке Unicode, поэтому пользователи могут давать осмысленные имена своим файлам. Linux, однако, не даёт полной поддержки для Unicode, поэтому если вы монтируете диск Windows *VFAT*, то увидите, что подобные имена состоят из вопросительных знаков и прочего мусора.

Здесь указан путь, как получить правильное преобразование:

1. Убедитесь, что ваше ядро скомпилировано с поддержкой *codepage* (в частности, с поддержкой

кодовых страниц *cp866* и *KOI8-R* — возможно, в виде модулей *ls /lib/modules/`uname -r`/fs/nls_**).

2. Добавьте следующие опции при выполнении *mount* команды *mount*:

3. *codepage=866,iocharset=koi8-r*

или пропишите в */etc/fstab/* примерно такую строчку:

/dev/hdc1 /Rack vfat iocharset=koi8-r,umask=0,showexec,codepage=866,quiet 0 0

для выяснения подробностей смотрите *man mount*.

Более подробно смотрите *Documentation/filesystems/vfat.txt* в исходниках ядра Linux.

Samba

Чтобы увидеть русские буквы в именах файлов на диске Samba, в файл */etc/smb.conf* следует добавить строчки:

```
[global]
character set = koi8-r
client code page = 866
preserve case = yes
short preserve case = yes
```

Первые две опции указывает внутреннюю кодировку имён файловой системы (*client code page*) и внешнюю кодировку пользователя (*character set*).

Следующие две опции указывают, что надо сохранять регистр длинных и коротких имён файлов соответственно.

Для Samba версии 3.x, которая для своих нужд использует *unicode*, следует использовать следующие настройки:

```
[global]
unix charset = koi8-r
dos charset = 866
```

Поддержка кириллицы в DOS эмуляторе

Это, видимо, единственный программный продукт, который требует присутствия CP866 кодировки. Основанием является то, что CP866- это родная кириллическая кодовая страница DOS. Большинство программ, работающих в DOS с кириллицей, ориентированы на CP866.

Для консольной версии (*dos*) вы должны загрузить только клавиатуру и экранный драйвер. Большинство драйверов DOS будет прекрасно работать. А. Беликов использует *rk* драйвер А. Страхова, который работает и для консоли, и X версии *dosemu*.

Другая альтернатива — это *g* драйвер Вадима Курлянда (последняя версия этого драйвера — 2.0). Он прекрасно настраивается и поддерживает много кодировок, в том числе CP866 и KOI8. Однако он, вероятно, не будет работать для X Window (по крайней мере, версия 1.14 не работает).

До сих пор остаётся популярной программа-кириллизатор *keyrus* (например на Украине). Кроме латиницы она умеет ещё две раскладки (их можно редактировать).

Эти русификаторы можно найти в большинстве Российских Internet архивов, например, в [ftp архиве Курчатовского Института Ядерной Физики](#).

Кроме того, есть патч в <http://www.ice.ru/~vitus/misc/>, который позволяет обойтись без русификатора клавиатуры. Этот патч рекомендуется в X-ах, так как позволяет избежать двух клавиш переключения: одной в DOS, другой — на остальном десктопе. Для работы с удалённым dosemu (по telnet/ssh/rsh) патч является необходимым условием.

Для X-овых версий dosemu вы должны установить соответствующий X шрифт с 866-ой кодировкой. Взять можно по адресам <http://www.ice.ru/~vitus/works/x11.html>, или <ftp.dosemu.org>, или взять у автора www.inp.nsk.su/~bolkhov.

Кроме шрифтов в кодировке *cp866*, там можно найти и другие кириллические шрифты. Например, *vga_bul8x16.pcf.gz* и *vga_bul.pcf.gz* имеют кодировку *МИК* (Болгария), *vga_rusci8x16.pcf.gz* имеет украинскую "досовскую" кодировку *ruscii*.

Для установки шрифтов для dosemu вы должны:

- Сделать этот шрифт доступным X-серверу — для этого следует выполнить процедуру, описанную в разделе ["Установка шрифтов для X Window"](#).
- Прикрутить этот шрифт к dosemu. Если шрифт только заменяет первоначальный шрифт *vga*, то он будет опознан по умолчанию. Иначе вы должны описать его в */etc/dosemu.conf*:
 - # Font to use (without filename extensions). For example:
 - X { updatefreq 8 title "MS DOS" icon_name "xdos" font "vga-alt"}

И в заключение вы должны загрузить драйвер клавиатуры. Обратите внимание, что вам не нужны экранные драйверы в X окне. Не все драйверы будут работать, но, по крайней мере, два из них будут: rk A. Страхова, и cyrkeyb Pete Kvitek; однако лучше воспользоваться патчем <http://www.ice.ru/~vitus/misc/>.

Глава 11. Кириллизация библиотек

Содержание

[GNU readline](#)

[Поддержка кириллицы в программах gtk](#)

В этом разделе описаны методы кириллизации программных библиотек.

GNU readline

В *~/.inputrc* поместить

```
set meta-flag on
set convert-meta off
set output-meta on
```

Во многих дистрибутивах версии *GNU readline* кроме файла *~/.inputrc* используют также и общесистемный конфигурационный файл */etc/inputrc* или */etc/Inutrc*. Это даёт возможность системному администратору сделать общесистемную "кириллизацию" сразу для всех пользователей. Если ваша система не использует общесистемный конфигурационный файл, то вы можете научить её это делать. Создайте файл */etc/inputrc*, а затем установите переменную окружения INPUTRC равную */etc/inputrc*.

Поддержка кириллицы в программах gtk

Если gtk не находит нужных шрифтов, он режет 8-й бит. Шрифты задаются в */etc/gtk/gtkrc.\$LANG*. Поэтому нужно создать файл *gtkrc.ru* или поправить существующий. Внутри должно быть примерно следующее (названия шрифтов, размеры и прочие атрибуты можно исправить по вкусу, но эти шрифты должны содержать кириллицу в koi8-г):

```
style "gtk-default-ru" {
    fontset = "-cronymx-helvetica-medium-r-normal-*_*_*_*_*-koi8-r\
              -cronymx-courier-medium-r-normal-*_*_*_*-*koi8-r\
              -cronymx-fixed-medium-r-normal--*_*-*_*-*-*koi8-r\
              -cronymx-times-medium-r-normal--*-*-*-*-*-*koi8-r"
}
class "GtkWidget" style "gtk-default-ru"
и скопировать этот файл в 'gtkrc.ru_RU.KOI8-R', или сделать symlink:
$ cd /etc/gtk/
$ ln -s gtkrc.ru gtkrc.ru_RU.KOI8-R
Подробности см. www.gtk.org, а также http://www.linux.org.ru/gnome/
```

gtk понимает национальные файлы gtkrc с именами типа gtkrc.ru_RU.koi8r (не gtkrc.ru_RU.KOI8-R). Это позволяет хранить файлы конфигурации для нескольких locales одновременно, при условии, что отсутствует gtkrc.ru.

Глава 12. Программы электронной почты и чтения новостей

Настройка программ электронной почты и чтения новостей для распознавания кириллицы не очень сложна, хотя вам следует знать основные принципы работы электронной почты и новостей.

Обычно программы Internet электронной почты состоят из двух частей: *MUA* (Mail User Agent — пользовательская программа электронной почты) и *MTA* (Mail Transfer Agent — программа рассылки электронной почты). *MUA* - это программа, которую вы используете для чтения, составления и посыпки электронных сообщений. Однако пользовательская программа электронной почты сама письма не посыпает; вместо этого она вызывает программу рассылки электронной почты, которая отвечает за посылку сообщения в соответствующем направлении, используя соответствующий протокол. Пример пользовательской программы электронной почты — Pine, программы рассылки электронной почты — qmail.

До недавнего времени и MTA, и MUA по умолчанию не были чисто 8-ми битными программами. Поэтому, когда вы посыпаете сообщение, скажем, из Америки в Россию, то вы никогда не будете уверены в том, что какая-нибудь промежуточная программа рассылки электронной почты не "откусит" в вашем сообщении 8-ой бит у каждой буквы в вашем сообщении. Поэтому был разработан ряд протоколов, которые позволяют закодировать любой тип данных с использованием только печатаемых символов из 7-ми битного ASCII. Эта группа протоколов зовётся *MIME* (MultiMedia Mail Encoding — мультимедийная кодировка электронных сообщений).

Так как MIME обычно сконфигурирован по умолчанию довольно правильно, мы не будем обсуждать его здесь. Мы поговорим о MIME, когда будет обсуждаться совместимость между русскими кодировками (См. раздел "[Совместимость данных в MIME формате](#)").

Мы начнём с настройки пользовательской программы электронной почты, потому что именно с этой программой вы работаете непосредственно. Ей нужно указать:

1. что письма будут содержать 8-битные символы,
2. с какой кодировкой вы работаете, и
3. с какой кодировкой письма будут посыпаться.

Кодировки нужно обозначать не произвольными именами, а согласно рекомендаций *IANA*.

Конечно, указывая кодировку для посыпаемых писем, вы должны действовать в соответствии с существующими стандартами. Стандарт для России: для электронной почты используется *KOI8-R*, именем этой кодировки согласно рекомендаций *IANA* является *koi8-r*. В Болгарии же для электронной почты используется кодировка *cp1251*, имя согласно *IANA* — *windows-1251*.

Дополнение: Довольно часто случается так, что те, кто посыпают вам письма, неправильно настраивают (или не настраивают вообще, а по умолчанию стоит чёрте-что — пример Outlook) свои почтовые программы. В заголовках писем вместо windows-1251 чаще всего стоит iso-8859-1, US-ASCII или UNKNOWN-8BIT. Для исправления получаемых писем создайте файл *~/.procmailrc* со следующим содержанием (но я предупреждаю вас, что после ошибки в *~/.procmailrc* вы можете потерять всю вашу почту):

```
:0 fHw
* ^Content-Type: *text/plain; *charset="?(iso-8859-1|US-ASCII|UNKNOWN-8BIT)"?
| formail -i "Content-Type: text/plain; charset=windows-1251"
```

Настройка пользовательской программы электронной почты (MUA)

Emacs

Если вы уже настроили *emacs* как таковой, то вам не нужно специально настраивать программу электронной почты, которая обеспечивается средствами Emacs. (Для настройки Emacs смотрите раздел "[Emacs и XEmacs](#)").

elm

Добавьте следующую запись в ваш `~/.elm/elmrc`:

```
CHARSET=koi8-r
```

pine

Добавьте следующую запись в `~/.pinerc` для персональной настройки или в `/usr/lib/pine.conf` для настройки всей системы (этот опцию можно настроить прямо из pine):

```
character-set=koi8-r
```

Вы можете также изменить настройку вашего Pine для того, чтобы предотвратить посылку письма в quoted-printable кодировке.

```
enable-8bit-nntp-posting  
enable-8bit-esmtp-negotiation
```

Кроме того, удостоверьтесь, что вы имеете свежую версию pine. До недавнего времени эта программа имела различные проблемы с русским кодировками.

Дополнение: Чтобы настроить перекодировку win в koi в pine в `.pinerc` следует прописать нечто вроде (это можно сделать и из самого pine):

```
display-filters=_CHARSET(iso8859-5)_ /usr/local/bin/icat,  
_CHARSET(utf-8)_ /usr/local/bin/ucat,  
_CHARSET(windows-1251)_ /usr/local/bin/wcat
```

Вместо icat, wcat и ucat можно написать, скажем, iconv -from что-надо -to что-надо, или recode. Главное — писать с полным путём, а то у неё паранойя (и это правильно).

mutt

Добавьте следующую запись в ваш `~/.muttrc`:

```
set charset=koi8-r  
set allow_8bit
```

Эта программа автоматически перекодирует получаемые письма, если они написаны не в стандартной кодировке. Кроме того, она умеет перекодировать отправляемые письма, если вы используете кодировку, отличную от стандартной для электронной почты в вашей стране — если вы хотите воспользоваться этой особенностью, то укажите это в `~/.muttrc`.

Например, если вы живёте в России и используете `cp1251`, то, чтобы ваши письма отправлялись в *KOI8-R*, поместите следующие три строки в `~/.muttrc`:

```
set charset=windows-1251  
set send_charset=koi8-r  
set allow_8bit
```

Если версия sendmail старше 8, то, вероятно, в `~/.muttrc` следует добавить директиву:

```
set use_8bitmime
```

Настройка программ рассылки электронной почты (MTA)

В "природе" существует несколько программ рассылки электронной почты (MTA). Это sendmail, qmail, smail, exim, и так далее.

sendmail

Начиная с версии 8, sendmail по умолчанию правильно обрабатывает 8ми битные данные. Если этого не происходит — проверьте опцию `EightBitMode` и опцию 7 в разделе `mailers` в вашем файле `/etc/sendmail.cf`.

Подробнее смотрите руководство "*Sendmail. Operation and Installation Guide*".

В данный момент sendmail больше не откусывает 8-мой бит. Однако он может **закодировать** 8-ми битные данные с помощью специального `base64` формата кодировки. Хотя большинство программ чтения электронных сообщений распознают и декодируют подобные сообщения обратно в 8ми битный текст, вы, вероятно, хотите посыпать просто текст и быть уверенным, что все работает нормально.

Настройка программы чтения новостей

Читать новостные группы можно с помощью *Emacs* (раздел "[Emacs и XEmacs](#)") или *pine* (раздел "[pine](#)"), но лучше воспользоваться специализированными программами — такими как *tin*.

В отличии от электронной почты в программах чтения новостей лучше не пользоваться *mime*, а писать сразу в *KOI8-R* — вас поймут и без *mime*.

tin

Если ваш *tin* работает, то есть вы можете читать/вводить кириллицу, то лучше ничего не менять, если же у вас возникают проблемы, то выполните следующие действия:

- Установите последнюю версию *tin* — можно пропустить
- Для включения возможности отображения кириллицы в файле конфигурации `~/.tin/tinrc` сделайте
 - `post_mime_encoding=8bit`
 - `mail_mime_encoding=8bit`
- Если вам необходимо включить *time* (смотрите раздел “[Совместимость данных в MIME формате](#)”), то установите переменную окружения `MM_CHARSET` равную `KOI8-R` и закомментируйте переменную `mm_charset` в файле конфигурации `~/.tin/tinrc`, или просто приравняйте `mm_charset` из `~/.tin/tinrc` `KOI8-R` (этота опция “перебивает” установку `MM_CHARSET`).
- `mm_charset=koi8-r`

gnus

Для начала надо настроить *MULE* (См. раздел “[Emacs и MULE](#)”).

Далее в `.emacs` добавить фразу:
`(gnus-mule-add-group "fido7" 'cyrillic-koi8)`

Глава 13. WWW и кириллизация

В отличие от программ электронной почты и чтения новостей, нет никаких стандартов для русской кодировки на WWW. Основная причина заключается в том, что компания Microsoft предоставляет программы создания Web сайтов, которые знают о существовании только одной русской кодировки `cpi251`, полностью игнорируя существование других стандартов.

Настройки, описанные здесь, очень примитивны. Они позволяют вам видеть страницы в `KOI8-R` кодировке. Если ситуация как-то изменится, я добавлю ещё информации.

lynx

Начиная с версии 2.6, вы можете выбирать соответствующее значение для дисплея — display Character set. *lynx* версии 2.8 позволяет смотреть страницы в любой кодировке, а не только в `KOI8-R`.

Для настройки следует выйти в меню 'options' и проверить следующие настройки:

```
display (C)haracter set : KOI8-R character set  
preferred document c(H)arset : koi8-r;q=1.0,windows-1251;q=0.5
```

Если русская (`KOI8-R`) страничка отображается *Lynx*ом неверно, то это означает, что страница неверно оформлена (см. раздел “[Как правильно подготовить свою WWW страничку](#)”). В последних версиях *Lynx* это обходится следующим образом: в настройках нажмите '^A' и выберите `koi8-r` (в основном отображаются заглавные буквы) или `windows-1251` (в основном прописные буквы). В старых версиях *Lynx* после этого следует нажать @ (включить raw режим). Лучше подстраховаться и запускать *lynx* с ключиком `--raw`.

links

Для просмотра WWW в текстовом режиме можно использовать [links](#) (опять наши братья чехи на высоте). Поддержка кириллицы уже встроена в программу — надо только установить кодировку терминала (в меню “Настройки”) и кодовую страницу документа (в меню “Вид”).

Mozilla

Возьмите уже русифицированную версию Mozilla, например с [ftp AltLinux](#)

Netscape Navigator

Лучше воспользоваться Mozilla, как более продвинутой с точки зрения кириллизации программой. Убедитесь, что вы используете Netscape версии 4.06 и выше. Начиная с этой версии, Netscape поддерживает кириллицу гораздо лучше.

Основные настройки

Наконец-то свершилось, теперь Netscape поддерживает русские кодировки правильным образом. Вам надо только правильно настроить шрифты *KOI8-R*. После этого, если например требуется показать документ, который использует *cp1251*, Netscape автоматически перекодирует весь документ в *KOI8-R* и правильно отобразит его, даже если у него нет доступа к *cp1251* шрифтам.

К сожалению, Netscape всегда будет перекодировать документы, использующие *cp1251*, в *KOI8-R*, даже если ему доступны также и шрифты для *ISO-8859-5*. Из за этой "глупости" вы сможете увидеть нерусские кириллические буквы только если в системе инсталлированы шрифты для *ISO-8859-5* и нет *ни одного* шрифта *KOI8-R*. Конечно, это решение неприемлемо, если вы используете *KOI8-R*. Мне неизвестно, как заставить Netscape показывать украинскую букву "г с чубом". Из стандартных кодировок она включена только в *cp1251* и *KOI8-U*.

Для того, чтобы правильно настроить *KOI8-R* в вашем Netscape, сделайте следующее:

- В меню Options/General Preferences/Fonts выберите Cyrillic (KOI8) кодировку.
- Выберите подходящие шрифты для этой кодировки — например, выберите Times(Cronyx) как пропорциональный шрифт, и Courier(Cronyx) — как fixed.
- Сохраните настройки.

ВНИМАНИЕ: Все больше и больше появляется WWW страниц, оформление которых сильно зависит от определённых шрифтов. Это в основном касается страниц, созданных под и для MS Windows. Я настоятельно рекомендую установить сервер шрифтов с поддержкой *TrueType*. С помощью него некоторые странички станут выглядеть гораздо лучше. Для выяснения подробностей смотрите раздел "[Использование шрифтов TrueType](#)".

Некоторые WWW странички в интернете правильно отвечают на запрос по поводу используемой при их создании кодировки. Другие требуют, чтобы вы выбрали кодировку сами. Для того, чтобы это сделать, выберите правильную опцию кодировки в меню Options/Document Encoding.

Netscape версии 4.08 правильно отображает элементы форм, использующих правильную кодировку (по крайней мере для большинства страниц, которые я видел). Более старые версии делают это неправильно. Однако, если по каким-то причинам вам надо использовать более старую версию, или правильное отображение форм по какой-либо причине не работает, то попробуйте сделать следующее:

1. Скопируйте базу данных установок Netscape (обычно *Netscape.ad*) в *~/Netscape*

2. В файле, установите следующую опцию:

3. **documentFonts.charset*iso8859-1: koi8-r*

Это вынудит все фреймы и элементы ввода использовать шрифты с кодировкой *koi8-r* вместо заданных по умолчанию, а следовательно, вы должны удостовериться в том, что вы уже установили такие шрифты (см. раздел "[Установка шрифтов для X Window](#)").

Плохая новость об использовании этого трюка заключается в том, что если вы загружаете документ, который должен отображаться с помощью шрифтов *iso-8859-1*, то вместо этого он отобразится с помощью *koi8* шрифта. Иногда такие документы выглядят хуже.

Принципиально не русифицируемые вебы: у обоих нетскейпов (3 и 4) будут трудности со страницами в кодировке, не соответствующей 'meta content-type' в заголовке (традиционная проблема серверов с выбором кодировок и авторов, пользующихся решениями от MS). Не лечится ничем, кроме хака бинарников. Что ещё хуже — если в документе явно указан шрифт, и такой шрифт, не дай Бог, у вас имеется (естественно, не русифицированный), то им все и будет нарисовано. Решений два: или не иметь в системе нерусифицированных шрифтов вовсе, или отключить в нетскейпе автоматическую загрузку изображений,

при этом "заодно", как ни странно, отключится и показ шрифтов, указанных в документе. Нажав Alt-I, вы получите возможность посмотреть картинки. Надеюсь, текст к этому моменту вы уже запомнили ;).
Если вам нужно чего-то ещё: Андрей А. Чернов — это человек, который знает больше о КОИ8 в общем и в netscape в частности, чем другие люди. Посетите его превосходную [KOI8 page](#) страницу, и скачайте заплату для файла ресурсов Netscape, которая заставляет Netscape говорить по-русски так хорошо, как это только возможно.

Как правильно подготовить свою WWW страничку

Этот раздел появился благодаря (фактически полностью состоит из) статьям Андрея А. Чернова. Подробную информацию по этому вопросу можно найти на страничке А.А. Чернова [KOI8-R - Russian Net Character Set](#).

Предисловие

Всемирное падение нравов началось с незначительной уступки пользователям. Как известно, в стандартах по HTML/HTTP четко сказано, что каждому документу должен соответствовать такой параметр, как MIME-имя кодировки, и есть только один случай, когда его можно не указывать — если документ составлен в кодировке *ISO 8859-1*. Это правило оказалось либо совершенно неизвестным, либо невыносимо сложным для неподготовленных пользователей, так что они предпочли проигнорировать его, и странички создавались просто набиванием некоторых кодов.

Далее, компания Netscape начал поступать поток претензий, что странички, набитые таким образом, не видны. Вместо того, чтобы отослать ламеров к документации, компания Netscape делает фатальный ход, минимально необходимый, чтобы все окончательно разрушить — вводят возможность выбора кодировки для страничек, где она не указана!

Естественно, после такого уже никто не считает нужным указывать кодировку документа; раньше останавливало хотя бы то обстоятельство, что были затруднения с просмотром кривых страничек, требующие такой достаточно сложной операции, как замена шрифта *ISO 8859-1* на шрифт используемой кодировки, теперь это препятствие было преодолено благодаря попустительству Netscape. Вслед за Netscape этот фатальный шаг делают и производители остальных браузеров, из соображений выживания в конкурентной борьбе. Вернуть все назад невозможно из-за тех же коммерческих соображений...

Тем самым, теперь выбор кодировки переносится с автоматического уровня на пользовательский, отсюда все эти надоедливые "Выберите вашу кодировку" и на страничках, и в браузерах.

В связи с общей коммерциализацией Интернет такие вещи, как соответствие стандартам, уже перестают быть аргументом, и продукты, еще следующие им, потихоньку вытесняются в маргинальные области. В общем, процесс повторяет происходящее в социуме, и будет продолжаться до тех пор, пока участников не начнет тошнить. Этот порог отодвигается все дальше и дальше благодаря удивительным способностям человеческого организма к адаптации...

ВНИМАНИЕ! Иногда я слышу безответственные "мнения" по поводу того, что указывать кодировку необязательно и не нужно: мол, пользователь у нас не ленивый, выберет вручную. При этом "советчики" злостно умалчивают о том, что надёжного автоматического определения кодировки не существует и не может существовать в принципе; так что вебмастера, не проставляющие кодировку, с какой-то вероятностью ставят свои странички под удар быть неверно истолкованными разного рода системами автоматической обработки текста, например, поисковыми машинами. Это все равно как вам бы предложили согласиться быть убитым с небольшой вероятностью, скажем 3%. И зачем вам это нужно?

Русские странички не нарушая стандартов - это просто!

Здесь приведена пошаговая инструкция правильной подготовки ваших русских страничек для WWW:

- Если у вас на страничках стоит выбор кодировок, выкиньте этот кусок HTML кода: он вам больше не понадобится — отныне все будет происходить автоматически.
- Если у вас странички не в кодировке *KOI8-R*, перекодируйте их туда каким-нибудь конвертером — смотрите раздел "[Символьная перекодировка](#)". Ничего страшного не произойдёт: если все дальнейшее вы сделаете правильно, то в таком виде их по-прежнему можно будет редактировать, скажем, в Netscape Composer без всякой заметной разницы. Мало того, в таком виде ваши странички будут одинаково хорошо видны для всех операционных систем без всяких переключателей кодировок, превратившихся в бич русского WWW.

- *Внимание!* Данный пункт можно пропустить, если на вашем сервере стоит Apache или какая-либо его разновидность (то есть в 90% случаев) — в этом случае необходимая информация может быть задана в настройках Apache. Если же вы предполагаете частые переезды с сервера на сервер с неизвестными HTTPD, следуйте указаниям пункта.
 - Проверьте самую первую строчку ваших русских HTML документов, она должна выглядеть примерно так:
 - <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
Если такой строчки нет, или номер после слова HTML меньше 4.0, добавьте её в самое начало. *Примечание для специалистов:* атрибут *LANG* определён только в HTML 4.0 и выше, так что указание DTD обязательно.
 - Замените следующий за ней таг <HTML> на таг <HTML LANG=ru>, это укажет, что ваш документ на русском языке, и облегчит жизнь поисковикам.
- Спросите вашего веб-мастера, что за штука стоит у вас в качестве веб-сервера, если он ответит:
 - что Apache или Apache/RUS: в самом верхнем каталоге, содержащем ваши странички, создайте файл с именем .htaccess, доступный всем на чтение с такими строчками:
 1. Для версий Apache, начиная с 1.3.12:
 2. AddDefaultCharset koi8-r
 3. AddLanguage ru .html .txt
 4. Для версий Apache, начиная с 1.3.10:
 5. AddCharset koi8-r .html .txt
 6. AddLanguage ru .html .txt
 7. Для более старых версий Apache (или если версия вам неизвестна):
 8. AddType "text/html; charset=koi8-r" .html
 9. AddType "text/plain; charset=koi8-r" .txt
 10. AddLanguage ru .html .txt

Если файл .htaccess уже есть, просто добавьте туда эти строчки.
Записанное в .htaccess распространяется на все подкаталоги вниз по дереву.
Подразумевается, что ваши HTML документы заканчиваются на .html; если они заканчиваются на .htm, поменяйте .html на .htm в предлагаемом образце.
При таких настройках не обязательно (но можно) указывать язык тагом <HTML LANG=ru>, как это было описано выше.
Указание языка внутри HTML документа имеет приоритет над указанием для сервера.
Так как для текстовых (*.txt) документов никак нельзя указать язык внутри документа, указание для сервера — это единственная альтернатива.
Если у вас на сервере стоит Apache/RUS, то рекомендуется отключить всякую автоматическую перекодировку, добавив в .htaccess директиву CharsetTurnOff On
(работает начиная с версии Apache/RUS PL20).
ПРЕДУПРЕЖДЕНИЕ: в последних версиях Apache обработка файлов .htaccess по умолчанию выключена в главной конфигурации (директивой AllowOverride None), т.е. ваши установки не будут иметь никакого эффекта. В этом случае попросите вебмастера включить такую обработку для вашего каталога, добавив директиву AllowOverride FileInfo
в соответствующую секцию <Directory ...> конфигурации Apache.
 - что NCSA: в каждом каталоге, содержащем ваши странички, заведите файл с именем .htaccess, доступный всем на чтение с такими строчками:

- AddType text/html; charset=koi8-r .html
 - AddType text/plain; charset=koi8-r .txt
- Если файл .htaccess уже есть, просто добавьте туда эти строчки.
Подразумевается, что ваши HTML документы заканчиваются на .html; если они заканчиваются на .htm, поменяйте конец строки на .htm соответственно.
- что CERN: для каждого вашего каталога с русскими страничками создайте подкаталог .web, в него положите файлы с именами, сконструированными добавлением суффикса .meta к именам в главном каталоге. Каталог .web и файлы в нем должны быть доступны всем на чтение. Например, файлу index.html соответствует файл .web/index.html.meta и т.д. Каждый такой файл должен содержать следующую строчку
 - Content-Type: text/html; charset=koi8-r
(для HTML файлов) или
Content-Type: text/plain; charset=koi8-r
(для текстовых файлов).
 - что какой-нибудь другой HTTP сервер: сперва попробуйте связать расширения .html и .txt по аналогии с решениями выше; если вы не знаете, как это сделать в вашем HTTP сервере, вам придётся редактировать непосредственно сами ваши странички: в секцию <HEAD> каждого HTML документа вставьте самой первой директивой следующий таг:
 - <META HTTP-EQUIV="Content-Type"
○ CONTENT="text/html; charset=koi8-r">
Т.е. каждый ваш документ должен будет выглядеть примерно так:

```
<HEAD>
<META HTTP-EQUIV="Content-Type"
CONTENT="text/html; charset=koi8-r">
<TITLE>Заголовок странички</TITLE>
</HEAD>
<BODY>
Содержимое странички
</BODY>
```

К сожалению, в этом случае невозможно указание кодировки для текстовых файлов (*.txt). На этом всё готово. При просмотре ваших страничек никто теперь не увидит мусора из умляутов вместо нормальных русских букв, и не придётся больше выбирать кодировки!

Проверка правильности кириллизации

Как проверить, удалось ли прописать кодировку и язык в HTTP заголовке?
Допустим, ваша страничка называется <http://www.yourserver.ru/yourpath/yourpage.html>

- Наберите: telnet www.yourserver.ru 80
- После соединения наберите:
- HEAD /yourpath/yourpage.html HTTP/1.0
Если ваша страничка находится на виртуальном сервере, то к приведённой выше строчке следует добавить Host: host.ru
Возможно, в некоторых системах придётся набирать вслепую.
- Нажмите Enter два раза. Вы увидите HTTP заголовок.
- Найдите в нем поле Content-Type и проверьте, что там стоит не просто text/html, а text/html; charset=koi8-r
- Если вы на сервере установили русский язык для ваших страничек, в заголовке должно присутствовать поле

- Content-Language: ru

Полезные советы

А совсем старые версии броузеров не показывают русских букв при таком способе, что делать?
Посоветовать пользователю сменить версию броузера на более новую. В качестве precedента, вспомните, что было с вводом прогрессивных JPEG, или анимированных GIF? Все поменяли версии на новые, никто не жаловался и не делал две копии страничек, одну с GIFами, а другую — без, а потом предлагал бы их выбрать...

А нельзя сделать так, чтобы 1) старые версии работали, 2) новые версии работали, и 3) чтобы пользователю было удобно?

Нельзя. Из этой тройки можно сделать вместе только два, на выбор. Из-за ненависти к пользователю у нас часто делают (1) чтобы старые версии работали и (2) чтобы новые версии работали (при этом стандартам не соответствует), я же предлагаю (3) чтобы пользователю было удобно и (2) чтобы новые версии работали (при этом стандартам соответствовало), а вы?

Глава 14. Кириллические шрифты

На настоящий момент на рынке кириллических шрифтов существует только один продавец — это ParaType. Остальные (в том числе и "свободные" сайты типа Vedi или Litera) предлагают в основном акцидентные шрифты, для набора текстов не пригодные. С ценами Paratype можно ознакомиться, естественно, на www.paratype.com. Однако при отборе шрифтов надо очень внимательно прочитать лицензию и договор^[8]. Кроме того, необходимо озабочиться грамотным подбором гарнитур для начертаний \rm, \sf, \sc, \tt так, чтобы они были совместны. Иначе можно было бы ограничиться "свободной Литературной" от ParaType, но вопрос: откуда взять замену всем остальным TeXовским шрифтам?! Так что лучше пользоваться стандартными шрифтами *lh* Ольги Лапко (раздел "[lh](#)"). Относительно недавно появился полностью свободный набор шрифтов Type1 *cm-super*, которые получены преобразованием из LaTeX-шрифтов (раздел "[cm-super](#)"). Возможно шрифты *cm-super* являются самым перспективным решением.

Шрифты Type1

Современные Type1 фонты поддерживают UNICODE и содержат русские символы с именами глифов типа "afii10049".

К сожалению, многие приложения видят только первые 256 символов Type1 шрифта (Latin1 или ISO_8859-1). Именно для таких приложений нужны "отхаканные" шрифты, где в диапазоне 0-255 расположены символы KOI8-R. Взять можно, например, с ftp.kapella.gpi.ru. Этот набор даже имеет необходимую часть для добавления к файлу Fontmap. Вы должны утянуть содержание каталога /pub/cyrillic/psfonts. В файле README описываются все необходимые подробности.

"Нормальные" Type1 шрифты есть в Adobe Type Manager (ATM) или в поставке OS/2.

На сайте www.paratype.com можно найти несколько бесплатных Type1 декоративных шрифтов — например, Pushkin.

В разделе [Глава 8. Вывод на печать и PostScript](#) можно найти дополнительную информацию по использованию Type1 шрифтов.

cm-super

Самый "молодой" из рассматриваемых пакетов. В будущем, вероятно, этот пакет будет базовым во всех дистрибутивах, так как является свободным.

Пакет CM-Super состоит из Type 1 шрифтов, которые были получены с помощью конвертации из EC/TC, ECC и LH шрифтов (Computer Modern font families). Пакет включает BCE стандартные глифы, входящие в T1, TS1, T2A, T2B, T2C и Adobe кодировки LaTeX.

Пакет CM-Super можно взять на любом CTAN архиве или на [ftp://ftp.vsu.ru/](http://ftp.vsu.ru/). Для установки следует внимательно прочитать README.

literat

В 1996 году компания ParaGraph International и Russian Foundation for Basic Research сделал шрифты literat доступными для бесплатного использования (подробнее смотрите файл licence из самого пакета). По отзывам, начертание этих шрифтов сильно отличается (причём далеко не в лучшую сторону) от того, что изначально называлось "гарнитура литературная"^[9] — шрифт явно не доделан^[10], но в 99% случаев результат вывода на печать вполне удовлетворителен. Из-за своей лицензии ошибки шрифтов нельзя поправить, поэтому для печати лучше использовать пакет cm-super (раздел "[cm-super](#)"). Сами шрифты доступны по адресу <ftp://vsu.ru>. Там лежат и сами шрифты, и все, что надо для их использования с LaTeX и PDFLaTeX. Для того, чтобы эти шрифты можно было использовать с LaTeXом, следует достать так же и пакет [literat-*tar.gz](#) — "виртуальные шрифты" для LaTeX (T2A, OT1, T1 и TS1 кодировки). Подробности по установке смотрите в файле описания этого пакета.

PSCYR

Набор шрифтов Type1 для работы с TeX/LaTeX.

Полное число представленных базовых начертаний на текущий момент более четырёх десятков. Пакет поддерживается Александром Лебедевым, автором словаря для проверки правописания, описанного в разделе "[Словарь Александра Лебедева](#)".

Структура коллекции

Коллекция состоит из двух архивных файлов (версия пакета может быть другой): PSCyr-0.4-type1.tar.gz, содержащего собственно начертания глифов в формате Type1 (файлы *.pfb) и их метрику и кернинг (файлы *.afm), и PSCyr-0.4-tex.tar.gz, содержащего tfm и vf-файлы, вектор кодировки T2A, определения шрифтов и стилевой файл для системы TeX, список шрифтов для dvips, а также документацию. Пакет состоит из двух частей:

1. PsCyr-type1-*tar.gz — содержит шрифты Type 1 и их метрики. В пакет включены гарнитуры: AcademyPSCyr, AntiquaPSCyr, HandbookPSCyr, CollegePSCyr, Lazurski, TimesNewRomanPSMT, ArialMT, TextbookPSCyr, MagazinePSCyr, CourierNewPSMT, KurierPSCyr, CooperPSCyr, AdvertisementPSCyr.
2. PsCyr-tex-*tar.gz — tfm и vf-файлы, вектор кодировки T2A, определения шрифтов и стилевой файл для системы TeX, список шрифтов для dvips, а также документацию.

Где можно взять коллекцию?

К первичному узлу PSCyr, можно получить доступ по [http](http://), или [ftp](ftp://) протоколу. Официальное зеркало находится по адресу: <ftp://vsu.ru>

Paradissa

[Басиль К. Малышев](#) в 1993 году выложил набор кириллических шрифтов при условии отсутствия модификации пакета (подробности смотрите в лицензии пакета). Сам пакет можно найти в любом архиве [CTAN](#) в директории fonts/cm/ps-type1/paradissa. Басиль К. Малышев известен как автор популярного коммерческого дистрибутива LaTeX под win32 BaKoMaTeX.

Пакет Paradissa имеет некоторые технические "опечатки", которые можно исправить с помощью простого скрипта `corr-cmc.sh', приведённого ниже. Скрипт использует утилиты t1disasm и t1asm из пакета t1utils. #!/bin/sh

```
for i in cmc*.pfb; do
    t1disasm $i | awk '
        /^\Subrs [0-9]* array$/ {s=1}
        /readonly def$/ && s {print "|";s=0;next}
        /^\.notdef {$/ {while ($0 !~ '/') +$0 {getline
            printf "/.notdef {\n\t0 0 hsbw\n\tendchar\n"
            {print}
        } | t1asm -b > .temp
        rm -f $i
        mv -f .temp `echo $i | sed 's,\(cmc.....\)\.pfb,\1.pfb,'
```

done

Из-за своей несвободной лицензии вместо шрифтов Paradissa лучше использовать пакет cm-super (раздел “[cm-super](#)”).

LucidaSO

Благодаря свободно распространяемому дистрибутиву StarOffice 5.2 (См. раздел “[StarOffice](#)”) стали доступны шрифты LucidaSO (версия 0.1 от 13 августа 2000).

Эти шрифты можно найти по адресу: [ftp://ftp.vsu.ru/pub/tex/font-packs/lucidaso/](http://ftp.vsu.ru/pub/tex/font-packs/lucidaso/). Там лежат и сами шрифты, и все, что надо для их использования с LaTeX и PDFLaTeX.

В пакет входит три шрифтовые гарнитуры: LucidaSans, LucidaType и LucidaBright (всего 32 шрифта type1). Поддерживаются следующие кодировки TeX: T1, TS1, OT1, T2A. Кириллические шрифты содержатся только в LucidaSans и LucidaType. Всего есть 70 начертаний.

Виртуальные шрифты созданы с использованием fontinst v1.914.

Для инсталляции шрифтов в TeTeX (См. раздел “[Кириллизация TeTeX](#)”) выполните следующие действия:

- распакуйте tar-gz файл lucidaso-type1-<date>.tar.gz в директорию *texmf*;
- распакуйте tar-gz файл lucidaso-tex-<date>.tar.gz в директорию *texmf*. *Внимание:* при распаковки могут затереться TFM и VF файлы Lucida шрифтов, которые являются частью дистрибутива TeTeX;
- добавьте в скрипт *updmap* запись для *lucidaso.map*;
- выполните *updmap* для перегенерации map файлов;
- для того, чтобы просмотреть документацию/примеры, загляните в директорию *texmf/doc/latex/lucidaso/*.

В случае возникновения комментариев, вопросов и "bug reports", свяжитесь с автором пакета Владимиром Воловичем по электронной почте <TeX@vvv.vsu.ru>.

dmtr-fonts

Это полностью свободный и сделанный с нуля шрифт Дмитрия Сорокина. Найти можно по адресу [dmtr-fonts-0.99-ipl1mdk.noarch.rpm](#)

sharatype

Свободные шрифты от Сергея Шарашкина можно найти по адресу ftp.ice.ru. Начиная с версии 2.0.4 в пакете появилось всё необходимое для использования этих шрифтов в TeX.

Шрифты для X Window

В этом разделе перечислены наиболее часто встречающиеся растровые шрифты.

Кроме растровых шрифтов, в X Window можно использовать Type1 Postscript шрифты (раздел “[Шрифты Type1](#)”). Правда, Type1 шрифты не стоит использовать как терминальные (обычно в этих шрифтах отсутствуют необходимые для этого символы), но в некоторых случаях они предпочтительнее, так как являются масштабируемыми.

Cyr-RFX

Наиболее эргономичные, на текущие момент, кириллические шрифты для X Window — это шрифты из проекта Cyr-RFX Дмитрия Болховитянова: <http://www.inp.nsk.su/~bolkhov/files/fonts/cyr-rfx/>. Его шрифты предлагаются для *всех* используемых кодировок кириллицы.

Настоятельно рекомендуется для использования в качестве базового набора.

Cronyx

В X Window, ещё в конце 1995 года, в качестве стандартного пакета был включён набор *KOI8-R* шрифтов, разработанных [Cronyx](#). Эти шрифты являются также частью XFree86.

Пакет для XFree86 можно найти на любом FTP архиве, где лежит дистрибутив X Window, например, непосредственно на [официальном XFree86 FTP архиве](#).

Имеется также более старое решение, например, пакет *Vakufonts*, созданный [Сергеем Вакуленко](#), который стал основой для пакета, включённого в дистрибутив X Window. Очень важно, что имена шрифтов в старой коллекции не совпадают со стандартом полностью. Vakufonts, в общем, неплохой пакет, но иногда могут возникать различные сверхъестественные ошибки.

Шрифт *-cronyx-fixed-** с полным соответствием *koi8-g*, в том числе с рамками: <http://www.ice.ru/lists/cyrfonts>. Коллекция *Cronyx* имеет новые версии, а также переработанный (и по-моему лучший) вариант *by Serge Vakulenka*, который включает все славянские кириллические буквы, а также и некоторые "дореволюционные" русские буквы.

В качестве базового набора рекомендуется использовать шрифты *cug-rfx* (раздел [“Cyr-RFX”](#)).

xrus

Пакет *xrus*^[11] является модификацией изначальных шрифтов *Cronyx*. Его можно найти по адресу <ftp://kiae.su>. Последнюю версию пакета *xrus* можно найти на страницах Андрея Чернова: <http://koi8.pp.ru/>. Теперь их адрес x6rus-2.3.1-bin.tgz (и версия — 2.3.1). Возможно, Чернов будет менять их место, так что когда дело дойдёт до beta, можно спросить в списке *cugfonts* о предпочтительных адресах.

В качестве базового набора рекомендуется использовать шрифты *cug-rfx* (раздел [“Cyr-RFX”](#)).

European

Очень благородные "исовые" шрифты можно взять на любом зеркале GNU в каталоге *intlfonts*, например, [European.tar.gz \(131 кб\)](http://www.gnu.org/software/intl/fonts/European.tar.gz). В Debian они уже есть.

В качестве базового набора рекомендуется использовать шрифты *cug-rfx* (раздел [“Cyr-RFX”](#)).

хсуг

Пакет *хсуг* можно найти здесь: sawsoft.newmail.ru/LS/koi8-extended.html.

В качестве базового набора рекомендуется использовать шрифты *cug-rfx* (раздел [“Cyr-RFX”](#)).

Шрифты для TeX/LaTeX

При работе с TeX/LaTeX можно использовать Type1 шрифты — где их можно найти, описано в разделе [“Шрифты Type1”](#).

Внимание: При использовании Type1 шрифтов в тексте, насыщенном математическими символами, восприятие текста умеренно ухудшится. Математические СМ-шрифты не будут сочетаться с текстовыми шрифтами. Это утверждение не относится к шрифтам *cm-super* (раздел [“cm-super”](#)) и в меньшей степени к шрифтам *PsCyr* (раздел [“PSCYR”](#)).

В этом разделе описаны стандартные кириллические шрифты *LaTeX*, которых хватает в 99.99% случаев.

Ih

Лучшие кириллические шрифты всех времён и народов — кроме шуток.

Благодаря Ольге Лапко (Издательство Мир) мы имеем бесплатный шрифт, который позволяет оформлять научные тексты с помощью *LaTeX*, совершенно не заботясь о подборке шрифтов для формул/текста. Как правило, эти шрифты уже есть в дистрибутиве *LaTeX*. Эти шрифты развиваются — в будущем обещают новые начертания (относительно недавно обещали добавить *Concrete* — "Конкретная Математика").

Шрифты *Ih* всегда можно достать по следующим адресам:

- CTAN:fonts/cyrillic/Ih

- <ftp://ftp.vsu.ru/pub/tex/>
- <http://www.cemi.rssi.ru/cyrtug/files/cyremtex.98>
- <http://www.cemi.rssi.ru/cyrtug/eng/news.htm>

Начиная с версии 3.20 *LH* поддерживает стандартные кодировки LaTeX “[Пакет T2](#)”, кроме того, поддерживаются LCY, OT2 и др.

Шрифты TrueType

Сами шрифты ttf (Type42) впервые были предложены корпорацией Apple, но своего наивысшего расцвета достигли с помощью компании Microsoft.

Несколько самых распространённых шрифтов, которые входят в стандартную поставку M\$ Windows сама Microsoft выложила для бесплатного использования. Причина по которой эти шрифты распространялись через web страничку компании заключалась в переходе на Unicode^[12]. Летом 2002 года шрифты были убраны со web странички Microsoft.

На сегодня не существует сколько-нибудь полного набора ttf шрифтов, для качественной подготовки документов. То что есть годится только для небольших простых документов, типа короткой деловой записи. Шрифты от Microsoft являются наиболее распространёнными и поэтому если вам нужны именно ttf шрифты^[13], то другой альтернативы нет.

Лицензия с которой выкладывались ttf шрифты от Microsoft не запрещает их выкладывать на других web страничках, что и было сделано на сайте [Microsoft's TrueType core fonts on linux](#). Однако, как долго продлится подобное состояние не известно.

Шрифты лежат в архивах. По лицензии EULA их запрещено модифицировать, но не запрещено распаковывать. Для распаковки следует воспользоваться утилитой cabextract. В некоторых дистрибутивах^[14] существуют специальные пакеты—обвязки, с помощью которых вы сами можете собрать установочные пакеты для частного пользования.

Если есть возможность, то следует отказываться от использования ttf в пользу Type1, например, в пользу пакета cm-super (раздел “[cm-super](#)”). Есть гораздо более удобные способы обработки текста чем с помощью редакторах типа M\$ Word

^[8] То есть, какие реально у вас будут права, какие реально есть гарантии и какую реально поддержку и сопровождение обеспечивает Paratype.

^[9] Старые математические книжки 60-х годов.

^[10] Можно посмотреть и ужаснуться, как выглядит слово "Где".

^[11] Не путать это с программой xruskb, ранее известной как xrus

^[12] В ранних версиях Windows шрифты для русской версии поставлялись в кодировке CP1251 и, соответственно, документы набранные в старом окружении требуют корректного преобразования, для отображения в среде UNICODE.

^[13] Это может потребоваться, если вы пользуетесь программой OpenOffice для совместимости с M\$ Word

^[14] Например в дистрибутиве Debian, ALT.

Глава 15. Кириллизация дистрибутивов

В этом разделе кратко описываются способы настройки кириллицы, которые специфичны для различных дистрибутивов Unix. Способы разнятся не только от дистрибутива к дистрибутиву, но и от версии дистрибутива.

Внимание: Этот раздел не поддерживается и не развивается. Реальность может очень сильно отличаться от информации приведённой здесь.

Если локализация из коробки является одним из основных признаков по которому вы выбираете дистрибутив Linux^[15], то следует выбрать национальный дистрибутив, команда которого непосредственно участвует в процессе локализации. Для русскоязычного^[16] пользователя таким дистрибутивом является дистрибутивы от команды [AltLinux](#).

Для настройки кириллицы прежде всего полезно прочитать разделы [Глава 3. Консоль](#), [Глава 4. X Window](#), и только после этого, зная, что вы делаете, обратиться к подразделам этого параграфа.

SuSE

Дистрибутив SuSe— немецкий дистрибутив. Один из самых добрых дистрибутивов Linux. Кириллизация из коробки отсутствует, особенно это относится к самым последним версиям этого дистрибутива

Настройка текстового режима

Пример русификации консоли koi8^[17].

Запустите программу конфигурации и администрирования *yast* под суперпользователем.

Для выбора консольного шрифта в меню „*System administration*” следует выбрать пункт „*Set the console font*”. Я рекомендую выбрать какой-либо шрифт из серии „*Cyr_a8x*” (у меня стоит шрифт *Cyr_a8x16.pcfu.gz*) — сделайте этот шрифт „*As standart font*”.

Далее в меню „*System administration*” выберите пункт „*Change configuration file*”. Среди предложенных для изменения переменных нас интересуют:

- *CONSOLE_FONT*— консольный шрифт (как поменять эту переменную, было описано ранее; можно менять её и здесь),
- *CONSOLE_SCREENMAP*— установите переменную равной *koi2alt*,
- *CONSOLE_MAGIC*— установить равной „(K” (кавычки не писать),
- *KEYTABLE*— рекомендуется установить равной *ru3.map.gz* (переключатель правый *CP866*, раскладка стандартная), или *ru4.map.gz* (переключатель *CapsLock*, раскладка для Windows клавиатуры),
- *CONSOLE_UNICODEMAP*— *cyralt.uni*,
- *RC_LANG*— можно установить равной *ru_RU.KOI8-R*.

Все это можно сделать, подредактировав файл *rc.local* в */etc/*.

P.S. Если при установке указать, что ваш язык — „русский” (*LANGUAGE=russian*), то всё настроится автоматически (ну, почти).

X window

Делайте так же, как описано в разделе [Глава 3. Консоль](#)— настройка ничем от стандартной настройки не отличается.

RedHat 6.2

Довольно удачный дистрибутив лидера коробочных продаж— американской фирмы RedHat. В настоящее время не поддерживается.

Кириллизация консоли

Для управления шрифтами и клавиатурой в RedHat 6.2 используется пакет *console-tools* (Для более тщательной настройки проконсультируйтесь с разделом [“consoletools”](#))
Полная русификация консоли RedHat 6.2:

- В файл */etc/sysconfig/i18n* поместите следующие строчки:
 -
 - *LANG="ru_RU.KOI8-R"*
 - *SYSFONTACM=koi8-r*

- SYSFONT=UniCyr_8x16
- В файл /etc/sysconfig/keyboard добавьте:
-
- KEYTABLE="ru4"

Переключение по Caps Lock, раскладка Microsoft — для полного счастья в деле настройки следует добыть пакет *console-tools-cyrillic* (См. раздел “[Cyrillic console tools](#)”).

ВНИМАНИЕ: Очень полезная ссылка — /usr/doc/initscripts-4.xx.

X window

Делайте так же, как описано в разделе [Глава 4. X Window](#) — настройка ничем от стандартной настройки не отличается.

RedHat 5.2

Довольно старый дистрибутив. Один из первых массовых дистрибутивов.

Кириллизация консоли

Из чего же сделана русификация в RedHat? Как известно, в Linux существуют два конкурирующих пакета управления шрифтами и клавиатурой: *kbd* и *consoletools*. В RedHat 5.x для русификации консоли применялся пакет *kbd*. В RedHat 6.x применяется уже другой пакет — *consoletools*. Разницы между ними радикальной нет, но это, увы, делает несовместимым пакет *initscripts* для версий RedHat 5.x и 6.x.

Русификация (и вообще включение национальных установок) в Linux RedHat 5.2 осуществляется из пакета *initscripts*. Не будем вдаваться в подробности работы *SYSV init*, опишем краткий путь. Предполагается, что читатель данного текста умеет пользоваться пакетом *RPM*, который весьма важен в RedHat.

Клавиатура

Клавиатура в RedHat 5.2 на нижнем уровне управляетя утилитой *loadkeys* из пакета kbd-096a-8. Пакет стандартный, RedHat лишь добавил туда несколько новых раскладок.

Запускается *loadkeys* из /etc/rc.d/init.d/keytable, управляющий файл /etc/sysconfig/keyboard. Там прописывается единственная переменная KEYTABLE:

```
/etc/sysconfig/keyboard:  
=====  
KEYTABLE=ru1  
=====
```

Есть интерактивная утилита смены раскладки: *kbdconfig*, которая на самом деле лишь прописывает значение в /etc/sysconfig/keyboard.

Файлы клавиатурных раскладок лежат в /usr/lib/kbd/keymaps/.

Мне нравится раскладка ru_win_ctrl.map.gz от [IP Labs](#). Переключение по правому Ctrl, точка и запятая в русском регистре, как в Windows, т.е. рядом с правой кнопкой Shift. (На моей клавиатуре нарисовано именно так). Надо положить этот файл в /usr/lib/kbd/keymaps/i386/querty/, запустить *kbdconfig* и выбрать *ru_win_ctrl*. (Хотя и родная раскладка *ru1* тоже ничего — точка и запятая на Shift 6, Shift 7).

Включить новую раскладку на ходу можно, запустив /etc/rc.d/init.d/keytable start

Консоль

Консоль в RedHat 5.2 на нижнем уровне управляетя утилитами *setfont* и *mapscrn* из пакета kbd-096a-8. Все, что необходимо сделать, это выполнить следующие команды:

```
$ setfont Cyr_a8x16  
$ mapscrn koi2alt
```

Кроме того, необходимо переключить Charset G1 на вновь загруженный шрифт. Для этого нужно на все виртуальные консоли /dev/vty* вывести „магическую“ последовательность *ESC (K)*.

```
$ echo -ne "\033(K"  
К сожалению, с русификацией консоли в RedHat не все так прозрачно, как с клавиатурой. Отдельной „подсистемы“ для управления консолью нет. А если делать все по-RedHat'овски, то возникают проблемы.
```

RedHat рекомендует использовать „новый“ формат, с *UNICODE* таблицей:

```
$ setfont Cyr_a8x16 -u koi8
К сожалению, на „новом“ формате сползают рамки в mc и других программах. В RedHat 6 применяется
другой пакет: consoletools, и другая версия initscripts, там все сделано немного по-другому.
Но попробуем выкрутиться!
Все утилиты запускаются из файла /sbin/setsysfont (на самом деле это shell-скрипт), параметры для которого
живут в /etc/sysconfig/i18n. Не очень понятно, каково отношение между i18n и шрифтами, но тем не менее это
так.
```

```
/etc/sysconfig/i18n
=====
SYSFONT=
UNIMAP=
=====
```

Не очень понятно, откуда посыпать *ESC* (*K*). Можно посыпать его, например, из *~.profile*

X window

Делайте так же, как описано в разделе [Глава 4. X Window](#)— настройка ничем от стандартной настройки не отличается.

Debian

Пожалуй самый обширный и „фундаментальный“ на сегодня дистрибутив. Разработка построена таким образом, что проще влиться в неё, нежели организовать свою ветку, как это произошло с *RedHat* подобными дистрибутивами.

Этот раздел посвящён кириллизации *Debian 2.2*. Если какие-либо действия надо совершить для младших версий этого дистрибутива— это будет отмечено особо.

Настройка русской локали

В файле */etc/environment*. Исправьте *LANG=C* на *LANG=ru_RU.KOI8-R*. В этом случае у всех пользователей включая *root* сообщения будут выводиться на русском языке и они смогут вводить русские буквы.

Опционально: Для *root* выдача сообщений на русском не слишком удобна, поэтому добавляем в *~root/.bashrc* следующие строки:

```
LC_TIME=C; export LC_TIME
LC_MESSAGES=C; export LC_MESSAGES
```

Если версия вашего дистрибутива меньше 2.2, то дополнительно выполните команду:

```
# mv /usr/share/locale/ru_SU /usr/share/locale/ru_RU.KOI8-R
с привилегиями пользователя root.
```

Зачем же мы удаляем *ru_SU* и делаем локаль с таким большим именем? Все потому, что *SU*— это сокращение страны Судан, а эта локаль предназначена для людей из России, говорящих по-русски. Можно было бы оставить *ru_RU*, скажете вы. Но *ru_RU* уже существует, это локаль с кодировкой *ISO-8859-5*. Поэтому следует поставить *ru_RU.KOI8-R*.

Русификация текстового режима.

Если в системе все ещё инсталлирован пакет *kbd*, то замените его на *console-tools*. Вставьте в */etc/console-tools/config*:

```
SCREEN_FONT=Cyr_a8x16
APP_CHARSET_MAP=koi2alt.trans
Выполните команду:
```

```
cp /usr/share/keymaps/i386/qwerty/ru1.kmap.gz /etc/console-tools/default.kmap.gz
с привилегиями пользователя root.
```

Первый вопрос, который возникает— зачем грузить какой-то *APP_CHARSET_MAP*? Действительно, если загрузить шрифт *KOI8-R*, то можно будет не грузить карту перекодировки (*APP_CHARSET_MAP*). В этом случае вместо символов псевдографики будут появляться русские буквы 'Д', а иногда даже 'YO'. Чтобы побороть это, можно сделать свой тип терминала (в *debian* есть *linux-koi8*). Но гораздо проще и грамотнее поставить *cp866* шрифты и включить таблицу перекодировки. Последний шаг— копирование карты

клавиатуры. *ru1* рассчитана на 101-кнопочную клавиатуру без MicroSoft'овских клавиш и переключение по правому *Alt*.

Если версия вашего дистрибутива меньше 2.2, то вместо console-tools вам придётся воспользоваться пакетом kbd — в этом случае настройки будут следующие:

Вставьте в /etc/kbd/config:

```
CONSOLE_FONT=Cyr_a8x16
CONSOLE_MAP=koi2alt.
```

А так же необходимо добавить следующие строки для использования русских шрифтов и маппинга на всех виртуальных консолях:

```
SCREEN_FONT_vc1=Cyr_a8x16
SCREEN_FONT_vc2=Cyr_a8x16
...
SCREEN_FONT_vc6=Cyr_a8x16
APP_CHARSET_MAPvc1=koi2alt.trans
APP_CHARSET_MAPvc2=koi2alt.trans
...
APP_CHARSET_MAPvc6=koi2alt.trans
```

Выполните команды:

```
# cd /etc/console-tools
# mv default.kmap.gz default.kmap.gz.orig
# ln -s /usr/share/keymaps/i386/qwerty/ru1.kmap.gz default.kmap.gz
```

с привилегиями пользователя root.

Русификация X Window System

Установите пакет *xfonts-cyrillic* со второго CD. Затем выполните команду

```
# ln -s /usr/X11R6/lib/X11/fonts/cyrillic /usr/X11R6/lib/X11/fonts/local
```

Теперь загрузите *XF86Setup* и скажите, что хотите Russian клавиатуру. Если вы не хотите пользоваться *xdm*, то в /etc/inittab измените id:2:initdefault: на id:1:initdefault:^[18].

Если же вы хотите воспользоваться *xdm*, то, вам надо будет вставить строчку

```
. /etc/environment
export LANG
```

в начале инициализационного сценария *xdm* (/etc/init.d/xdm). Затем в файле /etc/X11/xdm/xdm-config в конце вставьте строчку

```
DisplayManager.exportList: LANG
```

Зачем создавать символическую ссылку на /usr/X11R6/lib/X11/fonts/cyrillic? Дело в том, что *XF86Setup*, если видит каталог /usr/X11R6/lib/X11/fonts/local, то ставит его на первое место в секции „Files” файла /etc/X11/XF86Config. Зачем делать такие сложные выкрутасы с *xdm*? Чтобы *xdm* воспринял русскую локаль. Если вы хотите установить шрифты *Cyr-RFX-koi8-ru* (См. раздел [“Cyr-RFX”](#) — для кириллицы отличной от русской), то у вас возникнут проблемы с приложениями *gtk* — как с эти бороться описано в разделе [“Поддержка кириллицы в программах gtk”](#)).

Slackware

Старейший дистрибутив Linux. Своей похожестью на набор „Сделай сам” привлекает массу почитателей. К сожалению, не имеет кириллизации из коробки.

Этот раздел является сильно сокращённой версией (только кириллизация) Slackware-7-INSTALL-RUS.HOWTO (c) 1999 Вячеслава Калошина (multik@sw.ru, multik@istu.edu)

Русификация текстового режима.

Для кириллизации следует выполнить следующие действия.

- Переименовать файл /etc/rc.d/rc.font.sample в rc.font и добавить следующее содержание:
 -
 - setfont Cyr_a8x16
 - mapscrn koi2alt
 - for n in 1 2 3 4 5 6; do

- echo -ne "\033(K" > /dev/tty\$
• done
• loadkeys ru.map
таким образом, появятся русские буквы в консоли. Шрифты переключаются правым Ctrl.
P.S. В некоторых версиях дистрибутива этот метод не работает, так как после обработки rc.font грузится американская раскладка, которая „забивает” кириллицу. В этом случае перечисленные выше фразы следует поместить в rc.local
- Положить файл .inputrc следующего содержания в каталог /etc/skel:


```
•
• $if Bash
•     set convert-meta off
•     set output-meta on
•     set input-meta on
• $endif
```

Примечание к Slackware 8.0: В этой версии больше нет необходимости указывать что-либо в файле .inputrc, если вы правильно настроили локаль.
- В /etc/profile следует добавить:


```
•
• export LANG=ru_RU.KOI8-R
• export NLSPATH=/usr/share/locale/ru_RU/
• export LESSCHARSET=koi8-r
```
- При этом надо положить русскую локаль (её можно взять в других дистрибутивах, или скачать и развернуть [отсюда](#)) в /usr/share/locale/ru_RU/.
- от root'a выполнить следующие команды:


```
•
• cd /usr/share/locale
•
• localedef -c -f /usr/share/i18n/charmaps/KOI8-R -i \\
  /usr/share/i18n/locales/ru_RU \\
  ru_RU.KOI8-R
•
• ln -s ru_RU.koi8r ru_RU.KOI8-R
  символами \\ обозначено продолжение строки— эти строчки надо вводить в одну строку.
```
- в том же каталоге следует поправить файл locale.alias— добавить строчку:


```
•
• ru ru_RU.KOI8-R
```

Все, Slackware на нижнем уровне русифицирована.

Русификация X Window System

Делайте так же, как описано в разделе [Глава 4. X Window](#)— настройка ничем от стандартной настройки не отличается.
Но можно поступить, как советует Вячеслав Калошин (хотя этот способ не очень-то и правильный, но тоже работает):
Скопируйте файл [Xmodmap](#) в свой домашний каталог. Этот файл обеспечит ввод русских букв по правой кнопке Alt.
Проверьте в /etc/XF86Config секцию "Keyboard", добавьте туда строчку XkbDisable, иначе вышеприведённый способ русификации не будет работать.

[\[15\]](#) Например, если вы выбираете дистрибутив для домашнего компьютера

[\[16\]](#) А так же, вероятно, для украиноязычного и беларусоязычного.

[\[17\]](#) Проверено для версии 6.3

[\[18\]](#) Вместо правки /etc/inittab лучше поправить /etc/rc2.d/: переместить из этой директории куда-нибудь в созданную /usr/local/sys-bakup линк на /etc/init.d/xdm (S99xdm)

Библиография

Местоположение HOWTO

[Авторская домашняя страница.](#)

Локализация и интернационализация

Алексей Воропаев. [Локализация, как она есть.](#) Страница о локализации (locale), и обо всем, что с этим связано.

[Рекомендации IANA.](#)

[Описание MIME.](#)

Полезные программы и пакеты

Антон Зиновьев. [Программы по работе с BDF-шрифтами.](#)

Дмитрий Болховитянов. [Шрифты из проекта Cyr-RFX.](#) Очень рекомендую.

[Русские шрифты для Ghostscript и StarOffice.](#)

Языковые ресурсы

[www.peoples.org.ru.](#) Подробная информация о языках имеющих хождение на территории России.

Различные кириллические ресурсы

[Cyrillic resources at KIAE.](#)

[Cyrillic resources at RELCOM.](#)

[Обзор по славянским и неславянским кириллицам.](#) Говорят очень хорошо. "Cyrillic Alphabets and Unicode" в .PS формате.

[Обзор по 8-ми битным кириллическим кодировкам.](#) Есть кое-какая информация про Unicode.

[A SURVEY OF THE USE OF MODERN CYRILLIC SCRIPT.](#)

[The Languages of Russia.](#) Multimedia Web Site.

[Cyrillic resources at FUNET.](#)

[Collection of Cyrillic resources.](#) Если у вас проблемы с обратными слэшами — просто удалите их.

[Полезные кириллические пакеты.](#)

Русские кириллические ссылки

Андрей Чернов. [KOI8.](#) KOI8: информация и настройка.

Андрей Чернов. [RFC 1489.](#)

[linux.org.ru Общая информация по Linux на русском.](#)

[linux.ru.net Общая информация по Linux на русском.](#)

[Русская команда переводов GNU.](#)

[Кириллизация GNOME.](#)

[Переводы kde.](#)

Ричард Столман. [Право на чтение.](#) Памфлет.

Беларуские кириллические ссылки

[Все о беларусизации Линукса.](#)

Болгарские кириллические ссылки

[Linux Society in Bulgaria](#). Эти парни прилагают большие усилия, чтобы "болгаризация" Линукса не делалась стихийно и необдуманно. Здесь вы найдёте много полезных ссылок и информацию. Если вы желаете начать какой-нибудь проект, то подпишитесь пожалуйста на список рассылки develop@linux.zonebg.com или dict@linux.zonebg.com, в зависимости от характера вашей работы. .

[linux.gyuet.ch](#). Здесь довольно успешно пытаются рассказать обо всем.

[linux-bg.exco.net](#). Информации, новости и др.

[ВРАТАТА](#). Новости, ссылки электронного журнала "Phreedom".

[SAdict](#). Англо-болгарский и болгаро-английский электронный словарь.

Антон Зиновьев. [Пакет для болгаризации Линукса](#).

Македонские кириллические ссылки

[Все для македонской кириллицы](#).

[Новости, македонская локализация KDE](#).

[Часто Задаваемые Вопросы о Македонии \(язык\)](#).

Сербские ссылки

[Сербская кириллица в Интернете](#).

Украинские кириллические ссылки

[The Linux Ukrainian HOWTO](#).

[Советы для использования KOI8-У](#).

[cad.ntu-kpi.kiev.ua](#). Очень хорошая страничка, описания украинских кодировок и настройки программ, куча ссылок, есть koi8-и TTF шрифты.

[www.linux.org.ua](#). Здесь не стоит читать про украинизацию, но есть информация про перевод софта на украинский.

[www.net.ua/koi8-u](#). Говорят что информация здесь надёжна, ибо net.ua.

Andrew Porokhnyak. [www.ic-chernobyl.kiev.ua/~porokh/](#). Подробная информация о поддержке кодировки RUSCII (CP-1125) в разных программах, а также об украинских кодировках вообще. <aop@olmi.kiev.ua> <2.463/327@fidonet>.

[http://www.brama.com/compute/](#). Общее описание украинских кодировок, софт для Unix, Windows, Dos (шрифты, словари для ispell, переключалки клавиатуры)..

Азиатские кириллические ссылки

[ftp.logic.ru](#). Здесь выложены: раскладки клавиатуры для башкирского и кыргызского алфавитов; locale ru_RU с символами PT154;pt154.def из кириллического пакета LaTeX (он не входит в бинарные дистрибутивы teTeX).

Монгольские кириллические ссылки

[Mongolian HOWTO](#). Раздел монгольской письменности.

Предметный указатель

Symbols

Азиатские языки, [Азиатские языки](#)

Алфавит

русский, [Русский язык](#)

Беларусь, [Беларуский язык](#)

Болгария, [Болгарский язык](#)

Буквы

кириллица, [Символы и кодировки](#)

Графические редакторы

tgif, [tgif](#)

xfig, [xfig](#)

Интернационализация, [Локализация и Интернационализация](#), [Интернационализация](#)

перевод

рекомендации, [Рекомендации по переводу сообщений программ](#)

участие в проекте, [Как принять участие в проекте перевода](#)

КОИ8, [Стандарт КОИ8](#), [Украинский язык](#), [Русский язык](#)

КОИ8-Р, [Русский язык](#)

КОИ8-У, [Украинский язык](#)

Кириллизация

библиотеки, [Кириллизация библиотек](#)

обратная связь, [Местоположение документа и обратная связь](#)

Кириллица

DOS эмулятор, [Поддержка кириллицы в DOS эмуляторе](#)

Encapsulated Postscript

psfrag, [Пакет psfrag \(eps\)](#)

кириллические шрифты, [Инкапсуляция шрифтов Type1 в Encapsulated PostScript \(eps\)](#)

GNU readline, [bash](#)

IrcII, [IrcII](#)

LaTeX, [Поддержка кириллицы в TeX/LaTeX](#)

T2, [Пакет T2](#)

citehack, [Кириллица и bibtex](#)

texinfo, [Texinfo](#)

справки, [Кириллица в справках](#)

стиль, [Стиль](#)

Midnight Commander, [mc \(The Midnight Commander\)](#)

Postscript, [Вывод на печать и PostScript](#), [Кириллица в PostScript](#)

GNU enscript, [GNU enscript](#)

a2ps, [A2ps конвертер](#)

ghostscript, [Ghostscript](#)

rxtt2pst, [rxtt2ps](#)

кириллические шрифты, [Добавление шрифтов кириллицы к Ghostscript](#)

преобразование из текстового формата, [Преобразование текста в PostScript](#)

преобразование текста в TeX, [Преобразование текста в TeX](#)

Samba, [Samba](#)

TeX, [Поддержка кириллицы в TeX/LaTeX](#)

Encapsulated PostScript, [Инкапсуляция шрифтов Type1 в Encapsulated PostScript \(eps\)](#)

T2, [Пакет T2](#)

Washington Cyrillic, [Работа с Washington Cyrillic](#)

psfrag, [Пакет psfrag \(eps\)](#)

teTeX, [Кириллизация teTeX](#)

WWW

подготовка, [Как правильно подготовить свою WWW страничку](#), [Русские странички не нарушая стандартов - это просто!](#)

проверка, [Проверка правильности кириллизации](#)

стандарт, [Русские странички не нарушая стандартов - это просто!](#)

WWW browsing, [WWW и кириллизация](#)

Mozilla, [Mozilla](#)

Netscape, [Netscape Navigator](#)

links, [links](#)

lynx, [lynx](#)

настройка Netscape, [Основные настройки](#)

X Window System, [X Window](#)

Xkb, [Настройка Xkb](#)

xmodmap, [xmodmap](#)

ввод, [Ввод с клавиатуры](#)

нерусская, [Нерусская кириллица для X Window](#)

шрифты, [Установка шрифтов для X Window](#)

шрифты TrueType, [Использование шрифтов TrueType](#)

gnus

настройка, [gnus](#)

ispell, [Проверка правописания на русском языке](#)

Книжник, [Словарь Константина Книжника](#)

Лебедев, [Словарь Александра Лебедева](#)

less, [less](#)

locale, [Locale](#)

использование, [Locale с точки зрения пользователя](#)
настройка, [Настройка локали](#)
программирование, [Locale зависимое программирование](#)
ls, [ls](#)
man, [man](#)
nroff, [nroff](#)
perl, [Поддержка кириллицы в perl](#)
locale, [Поддержка locale в perl](#)
символьная перекодировка, [Символьная перекодировка средствами perl](#)
printing
матричные принтеры, [Использование старого матричного принтера для печати кириллического текста](#)
rlogin, [rlogin](#)
shell, [Кириллизация shell](#)
shells
bash, [bash](#)
csh, [csh/tcsh](#)
ksh, [ksh](#)
tcsh, [csh/tcsh](#)
zsh, [zsh](#)
telnet, [telnet](#)
tin
настройка, [tin](#)
Символьная перекодировка, [Символьная перекодировка](#)
perl, [Символьная перекодировка средствами perl](#)
Файловая система MS Windows, [Кириллические имена файлов в файловой системе MS Windows](#)
азбука, [Кириллица](#)
библиография, [Библиография](#)
буквы, [Символы и кодировки](#)
введение, [Введение](#)
дистрибутив Linux, [Кириллизация дистрибутивов](#)
Debian, [Debian](#)
RedHat, [X window](#)
RedHat 5.2, [RedHat 5.2](#)
RedHat 6.2, [RedHat 6.2](#)
Slackware, [Slackware](#)
SuSE, [SuSE](#)
дистрибутив Unix, [Кириллизация дистрибутивов](#)
интернационализация, [Локализация и Интернационализация](#), [Интернационализация](#)
участие в проекте, [Как принять участие в проекте перевода](#)
история, [Кириллица](#)
кириллизация консоли
RedHat 5.2, [Кириллизация консоли](#)
RedHat 6.2, [Кириллизация консоли](#)
кодировка, [Символы и кодировки](#)
консольный драйвер, [Консоль](#)
локализация, [Локализация и Интернационализация](#)
настройка MTA, [Настройка программ рассылки электронной почты \(MTA\)](#)
sendmail, [sendmail](#)
настройка MUA, [Настройка пользовательской программы электронной почты \(MUA\)](#)
настройка X Window
RedHat 6.2, [X window](#)
Slackware, [Русификация X Window System](#)
SuSE, [X window](#)
настройка X window
RedHat, [X window](#)
настройка X windows
Debian, [Русификация X Window System](#)
настройка консоли, [Консоль](#)
consoletools, [consoletools](#)
cyrillic consoletools, [Cyrillic console tools](#)
kbd, [kbd](#)
настройка текстового режима
SuSE, [Настройка текстового режима](#)

новости, [Программы электронной почты и чтения новостей](#), [Настройка программы чтения новостей](#)
печатать, [Вывод на печать и PostScript](#)
проверка правописания, [Проверка правописания на русском языке](#)
программ, [Кириллизация программ](#)
редакторы, [Редактирование текста](#)
GNU emacs, [Emacs и XEmacs](#), [Emacs и MULE](#)
emacs, [Emacs и XEmacs](#), [Emacs и MULE](#)
joe, [joe](#)
vi, [vi](#)
vim, [vi](#)
xemacs, [Emacs и MULE](#)
русский алфавит, [Русский язык](#)
символы, [Символы и кодировки](#)
совместимость, [Совместимость](#)
MIME формат, [Совместимость данных в MIME формате](#)
текстовые процессоры, [Текстовые процессоры](#)
текстовый режим, [Консоль](#)
электронная почта, [Программы электронной почты и чтения новостей](#)
Emacs, [Emacs](#)
elm, [elm](#)
mutt, [mutt](#)
pine, [pine](#)

Кириллические шрифты, [Кириллические шрифты](#)
LaTeX, [Шрифты для TeX/LaTeX](#)
LucidaSO, [LucidaSO](#)
Paradissa, [Paradissa](#)
PsCyr, [PSCYR](#)
cm-super, [cm-super](#)
lh, [lh](#)
literat, [literat](#)
sharatype, [sharatype](#)
Microsoft's TrueType core fonts, [Шрифты TrueType](#)
TeX, [Шрифты для TeX/LaTeX](#)
LucidaSO, [LucidaSO](#)
Paradissa, [Paradissa](#)
PsCyr, [PSCYR](#)
cm-super, [cm-super](#)
lh, [lh](#)
literat, [literat](#)
sharatype, [sharatype](#)
TrueType, [Шрифты TrueType](#)
Type1, [Шрифты Type1](#)
LucidaSO, [LucidaSO](#)
Paradissa, [Paradissa](#)
PsCyr, [PSCYR](#)
cm-super, [cm-super](#)
dmtr-fonts, [dmtr-fonts](#)
literat, [literat](#)
sharatype, [sharatype](#)
X Window, [Шрифты для X Window](#)
Cronyx, [Cronyx](#)
Cyr-RFX, [Cyr-RFX](#)
European, [European](#)
xcyr, [xcyr](#)
xrus, [xrus](#)
ttf, [Шрифты TrueType](#)

Клавиатура
RedHat 5.2, [Клавиатура](#)

Кодировка
ASCII, [Стандарт ASCII](#)
CP1251, [Стандарт CP1251](#), [Беларуский язык](#), [Болгарский язык](#)
CP866, [Стандарт CP866](#)
ISO, [Стандарт ISO](#), [Македонский язык](#)
UNICODE, [Стандарт UNICODE](#)

КОИ8, [Стандарт КОИ8](#), [Украинский язык](#), [Русский язык](#)
КОИ8-Р, [Русский язык](#)
КОИ8-У, [Украинский язык](#)
кириллица, [Символы и кодировки](#)

Консоль, [Консоль](#)
Debian, [Русификация текстового режима](#).
RedHat 5.2, [Консоль](#)
Slackware, [Русификация текстового режима](#).
проверка кириллизации, [Проверка правильности кириллизации консоли](#)

Консольный драйвер, [Консоль](#)

Локализация, [Локализация и Интернационализация](#)

Македония, [Македонский язык](#)

Монголия, [Монгольский язык](#)

Нерусская кириллица, [Нерусская кириллица для X Window](#)

Пакет consoletools, [consoletools](#)
cyrillic, [Cyrillic console tools](#)

Пакет ghostscript, [Ghostscript](#)
кириллические шрифты, [Добавление шрифтов кириллицы к Ghostscript](#)
настройка, [Настройка ghostscript](#)

Пакет kbd, [kbd](#)

Перевод
рекомендации, [Рекомендации по переводу сообщений программ](#)
участие в проекте, [Как принять участие в проекте перевода](#)

Проверка кириллизации консоли, [Проверка правильности кириллизации консоли](#)

Россия, [Русский язык](#)

Русский алфавит, [Русский язык](#)

Сербия, [Сербский язык](#)

Символьная перекодировка, [Символьная перекодировка средствами perl](#), [Символьная перекодировка](#)

Словарь
Книжник, [Словарь Константина Книжника](#)
Лебедев, [Словарь Александра Лебедева](#)

Срыв консоли, [Что делать, если "слетела" консоль?](#)

Стандарт
ASCII, [Стандарт ASCII](#)
CP1251, [Стандарт CP1251](#)
CP866, [Стандарт CP866](#)
ISO, [Стандарт ISO](#)
UNICODE, [Стандарт UNICODE](#)
КОИ8, [Стандарт КОИ8](#)

Текстовые процессоры
OpenOffice, [OpenOffice](#)
StarOffice, [StarOffice](#)

Украина, [Украинский язык](#)

Улёт консоли, [Что делать, если "слетела" консоль?](#)

Языковая специфика, [Языковая специфика](#)

A

ASCII, [Стандарт ASCII](#)

G

GNU emacs, [Emacs и XEmacs](#)
MULE, [Emacs и MULE](#)
GNU readline, [bash](#), [GNU readline](#)
Gnome, [Поддержка кириллицы в программах gtk](#)
ghostscript, [Ghostscript](#)
кириллические шрифты, [Добавление шрифтов кириллицы к Ghostscript](#)
настройка, [Настройка ghostscript](#)

gnus
настройка, [gnus](#)

gtk, [Поддержка кириллицы в программах gtk](#)

H

Настройка консоли, [Консоль](#)
consoletools, [consoletools](#)
cyrillic, [Cyrillic console tools](#)
kbd, [kbd](#)

I

ISO, [Стандарт ISO](#), [Македонский язык](#)
IrcII, [IrcII](#)

M

M\$ Windows
cygwin, [cygwin](#)
Microsoft's TrueType core fonts, [Шрифты TrueType](#)
Midnight Commander, [mc \(The Midnight Commander\)](#)
man, [man](#)

O

OpenOffice, [OpenOffice](#)

S

Slackware, [Slackware](#)
консоль, [Русификация текстового режима.](#)
StarOffice, [StarOffice](#)
SuSE, [SuSE](#)
sharatype, [sharatype](#)
shell, [Кириллизация shell](#)
shells
bash, [bash](#)
csh, [csh/tcsh](#)
ksh, [ksh](#)
tcsh, [csh/tcsh](#)
zsh, [zsh](#)

T

TeX
T2, [Пакет T2](#)
psfrag, [Подписи в картинках](#)
TeX шрифты, [Шрифты для TeX/LaTeX](#)
LucidaSO, [LucidaSO](#)
Paradissa, [Paradissa](#)
PsCyr, [PSCYR](#)
cm-super, [cm-super](#)
lh, [lh](#)
literat, [literat](#)
sharatype, [sharatype](#)
TrueType, [Шрифты TrueType](#)
Type1, [Шрифты Type1](#)
LucidaSO, [LucidaSO](#)
Paradissa, [Paradissa](#)
PsCyr, [PSCYR](#)
cm-super, [cm-super](#)
dmtr-fonts, [dmtr-fonts](#)
literat, [literat](#)
sharatype, [sharatype](#)

tcsh, [csh/tcsh](#)

telnet, [telnet](#)

texinfo, [Texinfo](#)

tgif, [tgif](#)

tin

настройка, [tin](#)

ttf, [Шрифты TrueType](#)

U

UNICODE, [Стандарт UNICODE](#)

W

WWW

кириллица

стандарт, [Русские странички не нарушая стандартов - это просто!](#)

подготовка, [Как правильно подготовить свою WWW страничку](#), [Русские странички не нарушая стандартов - это просто!](#)

проверка, [Проверка правильности кириллизации](#)

X

X Window

переключение регистра, [Несколько практических рекомендаций.](#)

X Window System, [X Window](#)

Xkb, [Настройка Xkb](#)

xmodmap, [xmodmap](#)

Нерусская кириллица, [Нерусская кириллица для X Window](#)

шрифты, [Установка шрифтов для X Window](#)

X Window шрифты, [Шрифты для X Window](#)

Cronyx, [Cronyx](#)

Cyr-RFX, [Cyr-RFX](#)

European, [European](#)

хсур, [хсур](#)

xrus, [xrus](#)

X-ресурсы, [X-ресурсы](#)

XML, [Формат](#)

Xkb, [Настройка Xkb](#)

настройка, [Настройка Xkb](#)

практика, [Настройка](#)

рекомендации, [Несколько практических рекомендаций.](#)

теория, [Немного теории](#)

хсур, [хсур](#)

xemacs, [Emacs и XEmacs](#)

MULE, [Emacs и MULE](#)

xfig, [xfig](#)

xmodmap, [xmodmap](#)

xrus, [xrus](#)

B

bash, [bash](#)

C

CP1251, [Стандарт CP1251](#), [Беларуский язык](#), [Болгарский язык](#)

CP866, [Стандарт CP866](#)

Cronyx, [Cronyx](#)

Cyr-RFX, [Cyr-RFX](#)

Cyrill-HOWTO-russian

оформление, [Правила оформления документа](#)
citehack, [Кириллица и bibtex](#)
cm-super, [cm-super](#)
consoletools, [consoletools](#)
cyrillic, [Cyrillic console tools](#)
copyrights, [Благодарность и авторские права](#)
csh, [csh/tcsh](#)
cygwin, [cygwin](#)

D

Debian, [Debian](#)
Locale, [Настройка русской локали](#)
консоль, [Русификация текстового режима](#).
dmtr-fonts, [dmtr-fonts](#)

E

European, [European](#)
emacs, [Emacs и XEmacs](#)
MULE, [Emacs и MULE](#)
X-ресурсы, [X-ресурсы](#)

K

kbd, [kbd](#)
ksh, [ksh](#)

L

LaTeX
T2, [Пакет T2](#)
citehack, [Кириллица и bibtex](#)
texinfo, [Texinfo](#)
формулы, [Кириллические буквы в формулах](#)
citehack, [Кириллица и bibtex](#)
graphics, [Подписи в картинках](#)
texinfo, [Texinfo](#)
отступ, [Отступ для первого параграфа раздела](#)
перенос, [Перенос](#)
приложение, [Кириллица и нумерация приложений](#)
справки, [Кириллица в справках](#)
стиль, [Стиль](#)
формулы, [Кириллические буквы в формулах](#), [Перенос формул](#)

LaTeX шрифты, [Шрифты для TeX/LaTeX](#)

LucidaSO, [LucidaSO](#)
Paradissa, [Paradissa](#)
PsCyr, [PSCYR](#)
cm-super, [cm-super](#)
lh, [lh](#)
literat, [literat](#)
sharatype, [sharatype](#)

Locale

Debian, [Настройка русской локали](#)
LucidaSO, [LucidaSO](#)
less, [less](#)
lh, [lh](#)
literat, [literat](#)
locale, [Locale](#)
использование, [Locale с точки зрения пользователя](#)
настройка кириллицы, [Настройка локали](#)
программирование, [Locale зависимое программирование](#)

ls, [ls](#)

P

Paradissa, [Paradissa](#)

PsCyr, [PSCYR](#)

perl, [Поддержка кириллицы в perl](#)

locale, [Поддержка locale в perl](#)

символьная перекодировка, [Символьная перекодировка средствами perl](#)

psfrag, [Подписи в картинках](#), [Пакет psfrag \(eps\)](#)

R

RedHat 5.2, [RedHat 5.2](#)

клавиатура, [Клавиатура](#)

консоль, [Консоль](#)

RedHat 6.2, [RedHat 6.2](#), [X window](#)

rlogin, [rlogin](#)

Z

zsh, [zsh](#)