

The Linux Cyrillic HOWTO (rus)

Балдин Евгений, (E.M.Baldin@inp.nsk.su), Новосибирск, Россия (По мотивам Cyrillic HOWTO Александра Беликова (abel@bfr.co.il), Bloomberg L.P.) Версия 0.8 пока beta, 29 октября 2000

Здесь описывается, как настроить кириллицу для редактирования, просмотра и печати документов в Unix подобной системе (в основном это относится к OS Linux).

Содержание

1 Общие примечания	1
2 Теоретическое обоснование	7
3 Настройка текстового режима	10
4 X Window	13
5 Кириллизация shell	20
6 Редактирование текста	23
7 Использование кириллицы в программах электронной почты и чтения новостей	25
8 WWW и кириллизация	28
9 Текстовые процессоры	33
10 Вывод на печать и PostScript	39
11 Локализация и Интернационализация	43
12 Совместимость	48
13 Кириллизация дистрибутивов	52
14 Кириллический Офис	57
15 Кириллические шрифты	58
16 Библиография	62
17 Полезные ссылки	63

1 Общие примечания

1.1 Для тех, кто хочет что-то добавить

1.1.1 Общие слова

Как вы заметили, этот текст имеет версию 0.8 пока beta – этим я пытаюсь показать, что его не стоит рассматривать как конечный продукт; и верить всему, что здесь написано – совершенно не обязательно. Здесь много чего лишнего и много чего не хватает – ничего страшного, это вполне можно исправить.

Этот текст основан (на 40% состоит из) на переводе Cyrillic HOWTO А. Беликова версии 4.2 b2, сделанного мной уже больше года назад. По той причине, что развитие этого чрезвычайно полезного HOWTO остановилось, появилась идея (как я понял, она появилась не только у меня) продолжить его развитие.

Основные исправления сделаны с помощью текста *RU.LINUX Frequently Asked Questions* и множества других источников, которые я всегда пытаюсь указать. Если я кого-то забыл/пропустил – дайте мне знать и эта проблема будет мигом (ну, за какой-то вполне разумный промежуток времени) решена.

Чего не будет: я не буду поддерживать английскую версию этого документа. Следует добавить, что я не против английской версии как таковой – я обеими руками за – просто я этим заниматься не буду.

Чего я ожидаю: я ожидаю, что по прочтении этого документа вы сделаете некие выводы, оформите их в виде ЦУ и пошлете их мне – особенно я буду благодарен тем, кто отловит орфографические ошибки (их у меня много).

Последнюю версию этого документа можно всегда найти на моей страничке: <http://www.inp.nsk.su/baldin> в разделе Cyrillic HOWTO (rus) (beta версия).

1.2 Введение

В этом документе описываются приемы, которые нужны для редактирования, просмотра и печати документов с использованием кириллических символов (в основном это относится к русскому языку) под Linux. И хотя здесь предполагается, что вы используете Linux как операционную систему, большая часть информации одинаково применима и к другим разновидностям Unix. Я попытаюсь указывать на различия.

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 ее, следовательно, требует определенных 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 "всеобщей кириллизации". Я не собираюсь описывать здесь, 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.

ВНИМАНИЕ: X-Window, TeX и другие компоненты Linux – это сложные системы. Если вы делаете 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 дает вам понять всю серьезность процесса настройки такого типа. **Очень** рекомендуется сделать копии файлов конфигурации. Также неплохо бы иметь под рукой гуру (специалиста по Linux) – просто так, на всякий пожарный.

1.3 Местоположение документа и обратная связь

Последнюю версию этого документа всегда можно найти на моей страничке: <http://www.inp.nsk.su/baldin> в разделе Cyrillic HOWTO (rus) (beta версия).

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 *unmaintained*. Также его можно найти на различных FTP, имеющих отношение к Linux. Кроме того, этот документ

может быть включен в дистрибутив Linux как его часть.

Дословный перевод устаревшей версии можно найти на моей страничке: <http://www.inp.nsk.su/baldin> 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.

1.4 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: Антон Зиновьев, Edward C. Bailey, Bas V. Bakker, Алексей Богданов, Michael Van Canneyt, David Daves, Денис В. Дмитриенко, Vlad Harchev, Дмитрий Малыханов, Сергей О. Наумов, Илья К. Орехов, Winfried Truemper, Сергей Вакуленко, Александр Воробьев, Сергей Тарарышкин, Владимир Волович, Алексей Махоткин, Виктор Вагнер, Леон Кантер, Дмитрий Болховитянов, Александр Воронов, Иван Паскаль, Федор Подгорный 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 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 идет 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 допускается и поощряется; однако автора следует оповещать относительно любых подобных действий.

Все работы, использующие данный документ, включая любой документ серии Linux HOWTO, должны быть распространяться под этим же соглашением. То есть вы не можете налагать дополнительные ограничения на распространение своего продукта, основанного на данном HOWTO. Исключения к этим правилам 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 Vynum, 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 идет 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 by Evgueni 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.

1.5 Что нового

Исправлен раздел "Русификация teTeX- – взято из FAQ, поддерживаемого Александром Махоткиным (fido7.ru.tex). Остальные разделы о русификации teTeX, видимо, надо убрать. Федор Подгорный проверил орфографию и пунктуацию, но в свете последних изменений я добавил массу дополнительных ошибок – кто поможет?

1.5.1 0.1 alpha 23 июня 2000

Добавил новый раздел **13** (Кириллизация дистрибутивов), в нем добавил подраздел для русификации консоли дистрибутива SuSe – для затравки.

В раздел "Кириллизация дистрибутивов" добавил кириллизацию для консоли Linux RedHat 5.2 (файл [ru-rh52.html](#) – авторство определить затрудняюсь, сильно за это извиняюсь :)

Добавил новый раздел **14** (Кириллический офис) – в дальнейшем статьи про *LaTeX*, печать и т.д. станут подмножеством этого раздела. Там фактически ничего нет – посмотрите, если у кого что-либо есть, то добавьте – хотелось бы увидеть рецепты для tkPaint, xmgr и др.

Добавил новый раздел **15** (Кириллические шрифты) – по идее здесь следует собрать все, что известно про шрифты – где что лежит. Пока ничего нет.

Взял описание настройки локали для раздела **11.1.1** (Настройка локали) из FAQ для `ru.linux` – прежнее описание настройки поместил в раздел "Устаревшая процедура установки Locale- – следует ли ее удалить? Вопрос Александру Воропаеву: достаточно ли этого описания или надо включить текст с твоей странички?

Взял текст от Ивана Паскаля, урезал его – боюсь, что недостаточно хорошо – и добавил раздел **4.3.1** (Настройка Xkb).

1.5.2 0.2 все еще alpha 30 июня 2000

Отреагировал на заявку о том, что в фразе "не дай боже- слово "боже- пишется с большой буквы методом убирания этой фразы из текста.

Добавил пару слов про Unicode в разделе 2.2 (Символы и кодировки). У меня нет ничего про использование Unicode в среде Unix – у кого есть хоть какая-то полезная информация?

Добавил текст в раздел 11.1.1 (Настойка локали), взятый (выданный – взята глава "Как включить локализацию") из замечательного источника информации о локализации на русском *Локализация, как она есть*.

По мотивам www.sensi.org/~ak/linuxfaq/rulinux.faq.html

1. немного изменил раздел 12.2 (Символьная перекодировка);
2. немного поправил раздел 5.6 (mc);
3. добавил разделы 5.7 (nroff) и 5.8 (man);
4. исправил раздел 5.5 (less);
5. добавил раздел 5.12 (telnet);
6. добавил раздел 3.5 (Что делать, если слетела консоль?);
7. поправил раздел 6.4 (joe);
8. добавил разделы 15.2 (Где найти хорошие русские шрифты для иксов?) и 15.1 (Где взять русские Type1 ps шрифты в KOI8-R?);
9. пополнил раздел 6.2 (Emacs/Xemacs);
10. исправлен раздел 12.4 (Поддержка кириллицы в DOS эмуляторе);
11. пополнил раздел 8.3 (Netscape Navigator);
12. добавил раздел 12.5 (Поддержка кириллицы в программах gtk);
13. добавил раздел 5.11 (Samba);
14. дополнил раздел 9.2 (StarOffice).

Дополнил раздел 6.1 (Проверка правописания на русском языке).

Добавил раздел 9.1.1 (Пакет T2) – по мотивам readme из пакета t2.

Добавил раздел 15.3.1 (lh).

Дополнил раздел 14.2.1 (tgif).

Добавил раздел 15.1.1 (literat).

Добавил раздел 10.4 (Инкапсуляция шрифтов Type1 в файлы eps) – по мотивам FAQ от Махоткина.

Добавил раздел 15.1.2 (PSCYR).

Добавил раздел 15.1.3 (Paradissa).

Добавил раздел 15.4 (Шрифты ttf).

1.5.3 0.3 еще alpha 06.08.2000

Сменил имя Cyrillic HOWTO на Cyrillic HOWTO (rus).

Внес несколько исправлений от Антона Зиновиева (в Cyrillic HOWTO появилось несколько слов про Кириллицу вообще, а не только про русскую ее часть).

Добавил раздел 13.3 (Debian).

Добавил раздел 9.2.1 (StarOffice 5.2) – страничка Леона Кантера.

Всякие мелочи – elm, перекодировщики и т.д.

Проверил HOWTO на "вшивость" с помощью sgmlcheck – html сделать можно, а вот latex нельзя; можно ли как-нибудь *indx* в html запихнуть, кто знает?

PS, PDF и sgml версии можно найти на страничке <http://www.inp.nsk.su/baldin> в разделе Cyrillic HOWTO (rus) (alpha версия).

1.5.4 0.4 уже не alpha, 12 августа 2000

Распечатал версию 0.3 – поглядел на это дело и поменял немного структуру документа.

1.5.5 0.5 почти beta, 13 сентября 2000

Вернулся из научно-туристической командировки на Сицилию, взял себя в руки и выпустил обновления к предыдущей версии:

Правки от Антона Зиновьева (Болгария!!!) (благодаря ему переделке и исправлению подверглось примерно 15% всего HOWTO):

1. Куча мелких, но важных исправлений
2. Ссылки на шрифты для X-window
3. Добавлен раздел 4.4 (Нерусская кириллица для X Window)
4. Изменен раздел 5.5 (less)
5. Изменен раздел 7.3 (Использования кириллицы в программах электронной почты и чтения новостей)
6. Изменен раздел 8.3 (Netscape Navigator)
7. Добавлен раздел 10.5 (Ghostscript)
8. Исправлен раздел 13.3 (Debian) - адаптирован для версии 2.2.
9. Добавлен раздел 5.9 (ls)
10. Добавлен раздел 10.7 (Формат A4)
11. Добавлена куча ссылок
12. Исправлен раздел 3.1 (Консольный драйвер Linux)

Дополнен раздел 15 (Кириллические шрифты)

1.5.6 0.6 уже beta, 14 октября 2000

Стилистическая правка – то есть убрал и поправил все, что по прочтению уж очень резало глаза (естественно, резало мне).

Пара небольших, но важных исправления от Евгения Бырганова.

Добавил раздел 8.4 (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* <<http://koi8.pp.ru/>>.

Добавил раздел 13.4 (Slackware 7) Этот раздел является сильно сокращенной версией (только кириллизация) Slackware-7-INSTALL-RUS.HOWTO (с) 1999 Вячеслава Калошина (multik@sw.ru, multik@istu.edu).

Поправил раздел 8.1 (lynx).

Добавил раздел 7.3.1 (tin).

Добавил раздел 5.13 (IrcII).

Поправил раздел 7.1.3 (pine) – из Mandrake-Russian Frequently Asked Questions by Andrei Zaparii.

1.5.7 0.7 test1 пока beta, 19 октября 2000

Пришел diff с исправлением кое-каких ошибок от Алекса Отта – не перевелись еще богатыри :)

1.5.8 0.8 пока beta, 29 октябрь 2000

Андрей Соболев взялся за сложнейшую задачу – за вылавливание ошибок в тексте и частично это задачу выполнил (желающие проверить как он выполнил это задачу всячески приветствуются) Кроме этого по его наводке и кое-где ехидным комментариям Андрея поправлен раздел 6.3 (vim) и раздел 6.4 (joe), а так же еще пара-тройка мест.

Поставил версию sgmltools из дистрибутива Mandrake – теперь LaTeX генерится без проблем – надо только добавить опцию `-l russian`

Автоматизировал процесс получения выходных файлов с помощью Makefile, поэтому кроме sgml и html версий буду выкладывать дополнительно rtf и Postscript.

1.6 Что я хочу сделать

Разобраться со ссылками.

Научиться делать PS версию SGML документа без особого напряжения – пока это не получается. Кто поможет?

Разобраться с индексами – пока не получается.

Исправить орфографические ошибки – кто бы помог? Свяжитесь со мной.

Исправить смысловые ошибки. Если вы что-либо обнаружили, даже мелкую ошибку, – дайте мне знать, не полнитесь; я, в свою очередь, тоже постараюсь не лениться, хотя это довольно тяжело.

Разобраться с SGML.

Распространить этот HOWTO как можно шире.

2 Теоретическое обоснование

2.1 Настройка вашего окружения

Перед тем, как мы начнем настраивать различные части системы, нам надо настроить пару простых вещей. Большинство утилит, описываемых ниже, предполагают, что кириллические шрифты доступны и пользователь может вводить кириллические символы. Чтобы это действительно стало правдой, нам 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, а другой – графический, предоставляемый средствами X-Window. Оба этих режима требуют разной настройки, которая описывается ниже.

2.2 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*.

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 остается) *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*, которые имели специальные значения 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 занимал 7 младших значимых битов байта, тогда как

старший разряд использовался для целей управления (то есть для управления передачи в старых пакетах связи).

Концепция 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) – чистый ASCII, в то время как старший разряд добавляет еще 127 символов. Так как эта кодировка обратно совместима с ASCII (символ все еще занимает 8 бит и коды полностью соответствуют старому ASCII), эта кодировка стала широко использоваться.

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 содержит псевдографические символы. Программное обеспечение, которое не делает никаких предположений о символах, использующих 8-й 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 фидо буквы *H* 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, требующие какой-либо настройки, в этом документе не рассматриваются. Вместо этого я сфокусировал свои усилия на более сложных пакетах.

Для информации о том, как создавать *свое* чисто 8-ми битное программное обеспечение, смотрите раздел **11.1.2** (Locale зависимое программирование).

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

Мы уже упомянули стандарт *ISO 8859-5* как тот, в котором определяются кодировки для кириллицы. Но поскольку (как это часто случается со стандартами) он был разработан без принятия во внимание реальных процессов, проходящих в СССР (когда это еще было), 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* (см. ниже). 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 Югославии.

Другие стандарты для кириллицы включают так называемую **Alt** кодировку и кодовую страницу **Microsoft CP1251**. Вышеупомянутый **Alt** стандарт был разработан компанией Microsoft для MS-DOS довольно давно. Тогда еще слыхом не слыхивали про сети из IBM PC, и поэтому основное усилие состояло в том, чтобы сделать этот стандарт максимально совместимым со стандартом IBM. Поэтому **Alt** кодировка – это та самая IBM кодовая страница, где все специфические европейские символы в верхней половине 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. **Alt** стандарт все еще жив и чрезвычайно популярен в среде MS-DOS (система MS-DOS тоже жива, про OS/2 говорят, что она тоже жива – не знаю, не видел :). **Microsoft CP1251** кодовая страница – это попытка Microsoft придумать новый стандарт для кодировки кириллицы в Windows. Насколько я знаю, это не совместимо с чем-либо еще.

Однако, в некоторых странах – например, в Болгарии и, вероятно, в Беларуси, *cp1251* является стандартной кодировкой для *Linux* и для *e-mail*.

Ну и наконец; стандарт **KOI8**. В отличие от **Alt** и **CP1251**, он был разработан довольно давно для UNIX-машин. Так как говоря о *UNIX* мы подразумеваем *сеть*, то основной идеей при создании **KOI8** стандарта была идея об обеспечении перемещения кириллической информации по сети.

Еще раз вернемся в далекое 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 **KOI8** 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**, убирать восьмой бит каждого символа, *то мы все еще 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 еще и Болгарскую кириллицу. Но кодировка **KOI8** неизвестна и поэтому ее употребление в Болгарии бессмысленно. Даже самые ранние болгарские кириллизации были только для **cp1251**, хотя и делались на основе русских.

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*").

Также существуют и другие стандарты, которые отличны от ASCII и гораздо более адаптируемы. Наиболее известный из них – это **Unicode**. Этот стандарт пока не прижился в Unix вообще и в Linux в частности, но уже сейчас появляется все больше и больше программного обеспечения, которое имеет поддержку Unicode.

2.2.1 UNICODE

UNICODE – частичная реализация стандарта **ISO 10646**, совместим снизу вверх, т.е. первые 256 символов **UNICODE** = 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/> .

3 Настройка текстового режима

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

Для специфической информации, касающейся вашего дистрибутива, вам следует обратиться в раздел [13](#) (Кириллизация дистрибутивов).

3.1 Консольный драйвер Linux

Консольный драйвер Linux – прекрасный образец хорошо настраиваемого программного обеспечения. Он может менять как шрифты, так и раскладки клавиатуры. Для специфической информации, касающейся вашего дистрибутива, вам следует обратиться в раздел [13](#) (Кириллизация дистрибутивов).

Чтобы настроить консоль, вы можете воспользоваться одним из двух описанных ниже пакетов: *console-tools* <<http://www.multimania.com/ydirson/soft/lct/>> (См. раздел [3.2](#) (*consoletools*)) или *kbd* <<http://sunsite.unc.edu/pub/Linux/system/Keyboards>> (См. раздел [3.3](#) (*kbd*)).

В ранних версиях Linux альтернативы *kbd* не было, но в свежих дистрибутивах вместо *kbd* сейчас можно использовать пакет *consoletools*. 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 А, то консольный драйвер сначала ищет код А в таблице АСМ (*Application Charset Map*). Из таблицы он узнает, какой код Б согласно кодировки Unicode соответствует коду А. Вы определяете используемую вами кодировку посредством таблицы АСМ. Далее консольный драйвер ищет код Б в таблице SFM *Screen Font Map*. Таблица SFM определяет кодировку используемого шрифта подобно тому как АСМ определяет вашу кодировку. От SFM консольный драйвер узнает, какой символ активного шрифта имеет код Б согласно Unicode и нужно вывести на экран.

Существуют две важные модификации этой схемы.

Аппаратные ограничения современных компьютеров не позволяют использовать шрифты, имеющие более 512 символов. Отсюда очевидно, что иногда консольный драйвер не сможет найти код Б в SFM. Тогда ему на помощь приходит так называемая *fallback таблица*. Эта таблица определяет для кода Б возможные его аппроксимации Б1, Б2,... Например, если Б является кодом символа "левая двойная угловая кавычка", то возможно Б1 будет кодом символа "левая одинарная угловая кавычка", а Б2 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 начнет 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 АСМ 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, разберемся с 8-битовым режимом. Не всегда консольный драйвер *Linux* умел работать с *Unicode*. Конечно, тогда не существовали также и АСМ, 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 АСМ.

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 АСМ (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 АСМ, *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 АСМ 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` <[ftp://ftp.ice.ru/pub/fonts/linux/](http://ftp.ice.ru/pub/fonts/linux/)> Виктора Вагнера.

3.2 consoletools

Сам пакет можно найти по адресу `console-tools` <<http://www.multimania.com/ydirson/soft/lct/>>.

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* <<ftp://ftp.ice.ru/pub/fonts/linux/>> Виктора Вагнера.

3.3 kbd

Довольно много дистрибутивов *Linux* устанавливают *kbd* как обязательную часть системы. Сам пакет доступен по адресу *kbd* <<http://sunsite.unc.edu/pub/Linux/system/Keyboards/>>.

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*.

ВНИМАНИЕ: В старых версиях *Linux* запуск программы *setfont* под X-Window мог повесить систему. Сейчас в этом случае печатается сообщение об ошибке.

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 Alt кодировкой и *консольной раскладкой* (*console character map*). Это означает, что ваша консоль отображает Alt шрифты, но все кириллические символы, соответствующие **КОИ8-R** кодировки, соответственным образом отображаются в Alt и поэтому выглядят правильно. Преимущество этого метода заключается в том, что он позволяет использовать псевдографические символы кодировки Alt.

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 идет магическая последовательность
echo -ne "\033(K"
```

При русификации всех консолей при загрузке системы вместо строчки

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

следует использовать строчку

```
for i in 1 2 3 4 5 6 7; do echo -ne "\033(K" > /dev/tty$i; done
```

– это русифицирует все терминалы.

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

3.4 Проверка правильности кириллизации консоли

Теперь вы, вероятно, хотите проверить правильность кириллизации консоли.

Соответствующим образом настройте *bash* (раздел 5.1 (*bash*)) или *tcsch* (раздел 5.2 (*tcsch*)) – этот шаг *необходим*, перезагрузите его, затем нажмите правую клавишу `Control`. Удостоверитесь, что вы можете правильно печатать по-русски. Клавиша 'q' должна соответствовать "й", 'w' соответствует "ц", и т.д. (*qwerty*->*йцукен*).

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 передается управление X-Window. Следовательно, после того, как вы вышли из X (или переключаетесь на консоль), вы должны перезагрузить русский шрифт.

3.5 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, еще придется перезагрузить шрифты, но это смотря как консоль сорвало...)

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

```
$ stty sane
```

В RedHat все это делают `reset; setsysfont <ctrl-j>`

`reset` – из *ncurses*

См. 'man console_codes' и 'man stty'

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.

Для специфической информации, касающейся вашего дистрибутива, вам следует обратиться в раздел 13 (Кириллизация дистрибутивов).

4.1 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 *Cronux*, еще 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 придется устанавливать самим.

Подробнее о шрифтах для X-window и о том, где их можно найти, смотрите в разделе 15.2 (Шрифты для X-window).

Более современную коллекцию шрифтов для X windows можно взять у *Дмитрия Болховитянова* по адресу пакет *CYR_RFX* <http://www.inp.nsk.su/~bolkhov/files/fonts/cyr_rfx/>.

Поэтому 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

```
find / -name crox\*.pcf\*
```

для того, чтобы найти местоположение шрифтов кириллицы в системе. Вы должны будете сделать эти шрифты доступными для X сервера, как – я объясню 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 15.2 (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:

```
bdftopcf -o <font>.pcf <font>.bdf
```

Можете сжать их с помощью программы `gzip` (очень старые X сервера не поддерживает сжатые шрифты):

```
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.

4. 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:

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

5. 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 подробно смотрите `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 `~/.xsession`, для пользователей GNOME специальным файлом запуска обычно является `~/.gnomerc`). Добавьте следующие команды в этот файл:

```
xset +fp <новый каталог шрифтов>
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.

6. 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`.

4.2 Использование шрифтов TrueType

У технологии TrueType, взятой на вооружение операционными системами серии Windows, нет многих недостатков, которые присутствуют при работе со шрифтами стандартным образом у X. Кроме того, существует множество кириллических TrueType шрифтов.

Подробнее о том, где их можно найти, можно посмотреть в разделе 15.4 (Шрифты TrueType).

К счастью, эти шрифты можно использовать и в X Window. Существует TrueType сервер шрифтов *XFSFT* для X. Поддержка TrueType встроена во все современные дистрибутивы XFree86.

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

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

Далее приводится пример того, как можно установить TrueType шрифты. О том, как подсунуть их X Window – смотрите раздел 4.1 (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_i.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 одну вещь, имеющую отношение к кириллической настройке – 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 *xfonstsel* и поиграв с семейством шрифтов *ttf*.

4.3 Ввод с клавиатуры

В последних выпусках X (X11R61 и выше) имеются два "стандартных" способа ввода с клавиатуры: родной способ, работающий через утилиту *xmodmap* (см. раздел 4.3.2 (*xmodmap*)) (сейчас это способ стремительно устаревает, хотя является надежным и хорошо изученным), и новый – *Xkb* или *X KeyBoard* (раздел 4.3.1 (*Xkb*)).

4.3.1 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 ее значение всегда одно и то же, независимо от состояния модификаторов.

- *compat* (сокращенное от *compatibility*)

описывает "поведение- модификаторов. В *XKB* имеется несколько внутренних переменных, которые, в конечном счете, и 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, определенного 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 одноименным директориям в директории `{XROOT}/lib/X11/xkb` (в дальнейшем, я буду обозначать ее `{XKBROOT}`).

Весь набор компонентов, необходимых для настройки *XKB*, описывается в файле конфигурации X-сервера в секции *Keyboard*.

Настройка Чтобы настроить ХКВ, 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"
```

означает, что ХКВ должен в соответствии с правилами, описанными в файле `{XKBROOT}/rules/xfree86`, выбрать настройки для клавиатуры типа "pc104- (104 кнопки), русского алфавита (английский алфавит будет включен "по умолчанию"), вариант – "стандартный- (то есть, этот параметр можно было не писать). Если у вас "windows- клавиатура, 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".

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:switch* – переключение по правому Alt только на момент нажатия,
6. *ctrl:ctrl_ac* – CapsLock

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 ХКВ.

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 строки в файл конфигурации *XFree* в секции *Keyboard*.

Для смены кодировки можно воспользоваться любой из многочисленных программ переключения кодовых страниц через *xkb* – например, я пользуюсь программой *hxkb* Ивана Паскаля.

Offtopic: Если в файле `/usr/X11R6/lib/X11/xkb/compat/iso9995` поменять

```
indicator "Group2" {
```

на

```
indicator "Scroll Lock" {
```

то в ответ на переключение регистра будет загораться лампочка *Scroll Lock* на клавиатуре.

4.3.2 xmodmap

Описанное ниже относится к относительно старым версиям *X-Window* (кто бы сказал, к каким – заведомо к версиям, меньшим 3.3.3), в которых способ *Xkb* (раздел 4.3.1 (*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*.

4.4 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 кодировки. Для перекодировки BDF-шрифтов можно использовать программу *trbdf* из пакета *trscripts* <<ftp://lml.bas.bg/home/anton/linux/trscripts-1.2.tar.gz>>.

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

Если вам нужна поддержка *cp1251*, то вы столкнетесь с тем, что пока официально *XFree* не поддерживает эту кодировку. Имеются три решения:

1. Использовать дистрибутив, поддерживающий *cp1251* (напр. *Mandrake Linux*).
2. Патчить *XFree*. Патч поддержки *cp1251* <<ftp://ftp.logic.ru/pub/logic/linux/be-locale>> 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

```
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 ее 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 символических имен нужно использовать коды символов согласно `cp1251`. Например, для белорусской раскладки вместо

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

нужно использовать

```
key <AD01> { [ q, Q ],
              [ 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*.

5 Кириллизация shell

5.1 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*, используемой `bash`-ем. Поэтому совет о `.inputrc` кириллизует не только `bash`, но и многие другие программы (все, использующие *GNU readline*). Во многих дистрибутивах версии *GNU readline* кроме файла `~/.inputrc` используют также и общесистемный конфигурационный файл `/etc/inputrc` или `/etc/Inputrc`. Это дает возможность системному администратору сделать общесистемную "кириллизацию" сразу для всех пользователей. Если ваша система не использует общесистемный конфигурационный файл, то вы можете научить ее это делать. Создайте файл `/etc/inputrc`, а затем установите переменную окружения `INPUTRC` равную `/etc/inputrc`.

5.2 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
```

5.3 zsh

Сделайте то же самое, что делается для csh (5.2 (csh)). Файл инициализации в этом случае – .zshrc или /etc/zshrc.

5.4 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
```

5.5 less

Если локаль (см. раздел 11.1.1 (Настройка локали)) не настроена (а она обязана быть настроенной), 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 11.1.1 (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_STYPE,) и, как следствие, не будет icase search для русских букв.

5.6 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 5.7 (nroff).

В случае возникновения проблем при просмотре man статей – сверьтесь с разделом 5.8 (man).

Если у вас проблемы в виде уродливых оконных рамок, то проконсультируйтесь в разделе 3.1 (консоль Linux).

off-topic: если вы захотите, чтобы mc в окне Xterm был цветным, а не черно-белым, то установите переменную COLORTERM:

```
COLORTERM= ; export COLORTERM
```

5.7 nroff

Для того, чтобы через nroff можно было "пропустить- символы кириллицы, надо использовать его с ключем *-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 с Slackware'96). В файле /usr/lib/mc/mc.ext следует в строке вызова nroff' а изменить параметр вызова с *-Tascii* на *-Tlatin1*.

5.8 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*, переведенных на русский язык, но вот отобразить их не всегда удается.

Если у меня 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 5.5 (*less*)).

5.9 ls

Если локаль установлена неправильно (см. раздел 11.1.1 (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`.

5.10 rlogin

Удостоверитесь, 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'
```

5.11 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
```

5.12 telnet

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

```
DEFAULT set outbinary
```

Вы можете встретить проблемы при работе в кодировке `cp1251` – не передается 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` лучше использовать `ssh` (Secure Shell) – максимальная безопасность при минимуме затрат.

5.13 IrcII

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

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

6 Редактирование текста

В этом разделе я опишу настройку различных программ для работы с кириллическими текстами. Этот раздел не охватывает **текстовые процессоры**, настройка которых описывается позже (смотри раздел 9 (Текстовые процессоры))

6.1 Проверка правописания на русском языке

Для проверки правописания я использую программу GNU `ispell`. Она имеет очень гибкие настройки и возможности для расширения. Ее можно использовать при проверке правописания текстов, написанных на языках, отличных от английского, путем добавления новых **словарей**.

6.1.1 Словарь Константина Книжника

Константин Книжник создал *очень* хороший русский словарь для `ispell`. Вы можете найти его на *домашней страничке К.Книжника* <<http://www.ispras.ru/~knizhnik>>. В поставку словаря включен полезный скрипт, обеспечивающий инкрементный режим проверки правописания слов для `emacs`.

В идеальном случае, когда ваш `ispell` установлен правильно, вам надо только создать словарь, используя команды, обеспечиваемые файлом `Makefile` из дистрибутива. Однако с некой долей вероятности у вас возникнут проблемы с `ispell`, который откажется 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 инициализировать проверку правописания для русских текстов путем вызова программы `ispell` с опцией `'-d russian'`

Если вы используете Emacs (подробнее в разделе 6.2 (Emacs)), то, вероятно, вы не против добавить пункт в меню для проверки с русским словарем.

Константин Книжник советует выполнить следующие действия:

Чтобы иметь возможность переключить запускаемый EMACS'ом ISPELL на использование русского словаря, можно попробовать добавить в `.emacs` следующие строки

```
(setq ispell-dictionary-alist (cons
  ' ("russian"
    "[\xc0-\xdf\xe0-\xff]" "[^\xc0-\xdf\xe0-\xff]" "[']"
    nil ("-B" "-d" "russian") nil) ispell-dictionary-alist))
(setq ispell-menu-map-needed t)
(setq ispell-menu-map nil)
(load "ispell") ; reload ISPELL
```

Необходимые действия уже содержатся в `surconf.el` в директории `emacs`, поэтому в случае использования нижеследующего включать их в `.emacs` не нужно.

6.1.2 Словарь Александра Лебедева

Более правильным (с точки зрения способа создания), чем 6.1.1 (словарь Книжника), является "Словарь русского языка для `ispell`- *Александра Лебедева*."

Словарь постоянно совершенствуется, дополняется и корректируется. Последнюю версию словаря можно найти на сервере: `mch5.chem.msu.su` <<ftp://mch5.chem.msu.su/pub/russian/ispell/rus-ispell.tar.gz>>

Я использую именно этот словарь.

6.2 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. Настройка кириллицы требует некоего низкоуровневого (в духе Emacs Лиспа) "хака- и немного отличается для этих двух реализаций.

Минимальная поддержка кириллицы в GNU emacs (вам не нужно этого делать при настройке XEmacs) обеспечивается при выполнении следующих вызовов, добавленных в `.emacs` (при условии, что поддержка символов кириллицы установлена для консоли или для X Window соответственно):

```
(standard-display-european t)

(let ((m (current-input-mode)))
  (set-input-mode (car m) (nth 1 m) 1))
```

Это позволит вам видеть и вводить символы, находящиеся в верхней части ASCII таблицы.

Однако этого не достаточно. Emacs обрабатывает кириллические символы как специальные и, как следствие, не распознает границы русских слов и не делает различия между строчными и заглавными буквами. Чтобы обойти это, вам следует несколько модифицировать таблицы синтаксиса и регистра emacs:

```
(require 'case-table)

(let* ((ruc "\341\342\367\347\344\345\263\366\372\351\352\353\354\355\356\357\360\362\363\364\365\366\367\368\369\370\371\372\373\374\375\376\377\378\379\380\381\382\383\384\385\386\387\388\389\390\391\392\393\394\395\396\397\398\399\400\401\402\403\404\405\406\407\408\409\410\411\412\413\414\415\416\417\418\419\420\421\422\423\424\425\426\427\428\429\430\431\432\433\434\435\436\437\438\439\440\441\442\443\444\445\446\447\448\449\450\451\452\453\454\455\456\457\458\459\460\461\462\463\464\465\466\467\468\469\470\471\472\473\474\475\476\477\478\479\480\481\482\483\484\485\486\487\488\489\490\491\492\493\494\495\496\497\498\499\500\501\502\503\504\505\506\507\508\509\510\511\512\513\514\515\516\517\518\519\520\521\522\523\524\525\526\527\528\529\530\531\532\533\534\535\536\537\538\539\540\541\542\543\544\545\546\547\548\549\550\551\552\553\554\555\556\557\558\559\560\561\562\563\564\565\566\567\568\569\570\571\572\573\574\575\576\577\578\579\580\581\582\583\584\585\586\587\588\589\590\591\592\593\594\595\596\597\598\599\600\601\602\603\604\605\606\607\608\609\610\611\612\613\614\615\616\617\618\619\620\621\622\623\624\625\626\627\628\629\630\631\632\633\634\635\636\637\638\639\640\641\642\643\644\645\646\647\648\649\650\651\652\653\654\655\656\657\658\659\660\661\662\663\664\665\666\667\668\669\670\671\672\673\674\675\676\677\678\679\680\681\682\683\684\685\686\687\688\689\690\691\692\693\694\695\696\697\698\699\700\701\702\703\704\705\706\707\708\709\710\711\712\713\714\715\716\717\718\719\720\721\722\723\724\725\726\727\728\729\730\731\732\733\734\735\736\737\738\739\740\741\742\743\744\745\746\747\748\749\750\751\752\753\754\755\756\757\758\759\760\761\762\763\764\765\766\767\768\769\770\771\772\773\774\775\776\777\778\779\780\781\782\783\784\785\786\787\788\789\790\791\792\793\794\795\796\797\798\799\800\801\802\803\804\805\806\807\808\809\810\811\812\813\814\815\816\817\818\819\820\821\822\823\824\825\826\827\828\829\830\831\832\833\834\835\836\837\838\839\840\841\842\843\844\845\846\847\848\849\850\851\852\853\854\855\856\857\858\859\860\861\862\863\864\865\866\867\868\869\870\871\872\873\874\875\876\877\878\879\880\881\882\883\884\885\886\887\888\889\890\891\892\893\894\895\896\897\898\899\900\901\902\903\904\905\906\907\908\909\910\911\912\913\914\915\916\917\918\919\920\921\922\923\924\925\926\927\928\929\930\931\932\933\934\935\936\937\938\939\940\941\942\943\944\945\946\947\948\949\950\951\952\953\954\955\956\957\958\959\960\961\962\963\964\965\966\967\968\969\970\971\972\973\974\975\976\977\978\979\980\981\982\983\984\985\986\987\988\989\990\991\992\993\994\995\996\997\998\999\1000")
      (rlc "\301\302\327\307\304\305\243\326\332\311\312\313\314\315\316\317\320\322\323\324\325\326\327\328\329\330\331\332\333\334\335\336\337\338\339\340\341\342\343\344\345\346\347\348\349\350\351\352\353\354\355\356\357\358\359\360\361\362\363\364\365\366\367\368\369\370\371\372\373\374\375\376\377\378\379\380\381\382\383\384\385\386\387\388\389\390\391\392\393\394\395\396\397\398\399\400\401\402\403\404\405\406\407\408\409\410\411\412\413\414\415\416\417\418\419\420\421\422\423\424\425\426\427\428\429\430\431\432\433\434\435\436\437\438\439\440\441\442\443\444\445\446\447\448\449\450\451\452\453\454\455\456\457\458\459\460\461\462\463\464\465\466\467\468\469\470\471\472\473\474\475\476\477\478\479\480\481\482\483\484\485\486\487\488\489\490\491\492\493\494\495\496\497\498\499\500\501\502\503\504\505\506\507\508\509\510\511\512\513\514\515\516\517\518\519\520\521\522\523\524\525\526\527\528\529\530\531\532\533\534\535\536\537\538\539\540\541\542\543\544\545\546\547\548\549\550\551\552\553\554\555\556\557\558\559\560\561\562\563\564\565\566\567\568\569\570\571\572\573\574\575\576\577\578\579\580\581\582\583\584\585\586\587\588\589\590\591\592\593\594\595\596\597\598\599\600\601\602\603\604\605\606\607\608\609\610\611\612\613\614\615\616\617\618\619\620\621\622\623\624\625\626\627\628\629\630\631\632\633\634\635\636\637\638\639\640\641\642\643\644\645\646\647\648\649\650\651\652\653\654\655\656\657\658\659\660\661\662\663\664\665\666\667\668\669\670\671\672\673\674\675\676\677\678\679\680\681\682\683\684\685\686\687\688\689\690\691\692\693\694\695\696\697\698\699\700\701\702\703\704\705\706\707\708\709\710\711\712\713\714\715\716\717\718\719\720\721\722\723\724\725\726\727\728\729\730\731\732\733\734\735\736\737\738\739\740\741\742\743\744\745\746\747\748\749\750\751\752\753\754\755\756\757\758\759\760\761\762\763\764\765\766\767\768\769\770\771\772\773\774\775\776\777\778\779\780\781\782\783\784\785\786\787\788\789\790\791\792\793\794\795\796\797\798\799\800\801\802\803\804\805\806\807\808\809\810\811\812\813\814\815\816\817\818\819\820\821\822\823\824\825\826\827\828\829\830\831\832\833\834\835\836\837\838\839\840\841\842\843\844\845\846\847\848\849\850\851\852\853\854\855\856\857\858\859\860\861\862\863\864\865\866\867\868\869\870\871\872\873\874\875\876\877\878\879\880\881\882\883\884\885\886\887\888\889\890\891\892\893\894\895\896\897\898\899\900\901\902\903\904\905\906\907\908\909\910\911\912\913\914\915\916\917\918\919\920\921\922\923\924\925\926\927\928\929\930\931\932\933\934\935\936\937\938\939\940\941\942\943\944\945\946\947\948\949\950\951\952\953\954\955\956\957\958\959\960\961\962\963\964\965\966\967\968\969\970\971\972\973\974\975\976\977\978\979\980\981\982\983\984\985\986\987\988\989\990\991\992\993\994\995\996\997\998\999"))
      (i 0)
      (len (length ruc)))
  (while (< i len)
    (modify-syntax-entry (elt ruc i) "w ")
    (modify-syntax-entry (elt rlc i) "w ")
    (set-case-syntax-pair (elt ruc i) (elt rlc i) (standard-case-table))
    (setq i (+ i 1))))
```

Для этого был создан файл `rusup.el`, который содержит эти команды, также как и пару других удобных функций. Вы должны вызвать его в вашем `~/emacs`.

Обратите внимание на пакет `russian.el` <<http://www.math.uga.edu/~valery/russian.el>>, созданный Валерием Алексеевым (`valery@math.uga.edu`), который позволяет пользователю переключаться между кириллическим и стандартным модами ввода и преобразовывать текстовый буфер из одной русской кодировки в другую (это очень полезно для чтения текстов, импортированных из MS-DOS или Windows).

ВНИМАНИЕ: Помимо описанной выше настройки, существует альтернативный путь обучения обоим версиям emacs кириллице. Для этого используется MULE (MULtilanguage Emacs support). Для настройки кириллицы через MULE берем iso8859-5 шрифты (см. раздел 15.2 (Где взять шрифты для X-window?)), подключаем их к иксам, и выбираем в меню Mule нужную кодировку. Этот способ является более "идеологически- правильным.

Совет: При работе с X-window для Emacs, вероятно, лучше всего подойдут шрифты `lucidatypewriter` из коллекции шрифтов Болховитянова (См. раздел 15.2.2 (Суг-RFX)).

6.3 Работа с vi

Редактор vi (по крайней мере, его клон vim, присутствующий в большинстве дистрибутивов Linux) знает о существовании 8-ми битных символов. Это дает 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 встреченных мной *unix-like* системах `vim` правильно работал с русским языком после настройки `locale` (Как настроить локаль смотрите раздел 11.1.1 (Настройка локали)).

Для того, чтобы постоянно не переключаться в командном режиме между английскими и русскими буквами, необходимо выполнить мапирование 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=ë ` ,йq,цw,уе,кр,ет,ну,гу,ші,щo,зр,х[,ъ],фа,ыс,вd,af,пg,ph,оj,лk,дl,э',яз,чx,сс,л
```

Подробности – в `:help langmap`.

6.4 Редактирование текста в joe

Все проблемы `joe` можно решить, правильно настроив `locale` (См. раздел 11.1.1 (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 ее 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 (BCL 6.2, RH 5.2-6.1; FreeBSD) правильно распознаёт границы русских слов. А для преобразования регистра `man joe` советует `^K /` – фильтровать блок через `tr`, т.е. все сводится к работе `tr`.

7 Использование кириллицы в программах 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 отправки электронных сообщений. Однако пользовательская программа электронной почты сама письма не посылает; вместо этого она вызывает программу рассылки электронной почты, которая отвечает за отсылку 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 посылаете 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 зовется **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 12.1 (mime)).

Мы начнем 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 посылаться.

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

Конечно, указывая кодировку для посылаемых писем, вы должны действовать соответственно существующим стандартам. Стандарт для России: для электронной почты используется *KOI8-R*, именем этой кодировки согласно рекомендациям *IANA* является *koi8-r*. В Болгарии же для электронной почты используется кодировка *cp1251*, имя согласно *IANA* – *windows-1251*.

(Письма, которые получал (и посылал) Антон Зиновьев в Сербию и Македонию, использовали кодировку *ISO-8859-5*. Но потребители Windows вряд ли используют *ISO-8859-5* – так что какая кодировка является стандартом для этих стран, совершенно не понятно.)

Дополнение: Довольно часто случается так, что те, кто посылают вам письма, неправильно настраивают (или не настраивают вообще, а по умолчанию стоит черт-че – 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"
```

7.1 Настройка пользовательской программы электронной почты (MUA)

7.1.1 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 [6.2](#) (Emacs и XEmacs)).

7.1.2 elm

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

```
CHARSET=koi8-r
```

7.1.3 pine

Добавьте следующую запись в `~/pinerc` для персональной настройки или в `/usr/lib/pine.conf` для настройки всей системы (эту 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 путем, а то у нее параноя (и это правильно).

7.1.4 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 живете 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
```

7.2 Настройка вашей программы рассылки электронной почты (MTA)

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

7.2.1 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 посылать просто текст и быть уверенным, что все работает нормально.

7.2.2 Другие программы рассылки электронной почты

Я немного знаю о других программах рассылки почты. Если вы что-то знаете, что может быть важно для настройки кириллицы, то пожалуйста, сообщите мне.

7.3 Настройка программы чтения новостей

Читать новостные группы можно с помощью *Emacs* (раздел 6.2 (*Emacs* и *XEmacs*)) или *pine* (раздел 7.1.3 (*pine*)), но лучше воспользоваться специализированными программами – такими как *tin*.

7.3.1 tin

Установите переменную окружения `MM_CHARSET` равную *KOI8-R* и прокомментируйте переменную `mm_charset` в файле конфигурации `~/.tin/tinrc`, или просто приравняйте `mm_charset` из `~/.tin/tinrc` *KOI8-R* (эта опция "перебивает" установку `MM_CHARSET`).

8 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 *cp1251*, полностью игнорируя существование других стандартов.

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

8.1 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 8.4 (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).

8.2 links

Для просмотра WWW в текстовом режиме можно использовать *links* `<http://artax.karlin.mff.cuni.cz/~mikulas/vyplody/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").

8.3 Netscape Navigator

Убедитесь, что вы используете Netscape версии 4.06 и выше. Начиная с этой версии, Netscape поддерживает кириллицу гораздо лучше.

8.3.1 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 установлены 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 (KOI-8) 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 определенных 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 4.2 (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:

```
*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 4.1 (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 отображится с помощью koi8 шрифта. Иногда такие документы выглядят хуже.

Принципиально не русифицируемые вещи: 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 еще 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. Решение 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 еще: Андрей А. Чернов – это человек, который знает больше о KOI8 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.

8.4 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* <<http://koi8.pp.ru/>>.

8.4.1 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, компании 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 – вводит 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, ещё 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- злобно умалчивают о том, что надежного 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?

8.4.2 Русские странички не нарушая стандартов - это просто!

Здесь приведена по шаговая 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 [12.2](#) (Символьная перекодировка). Ничего страшного не произойдет: 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 ее в самое начало.

Примечание для специалистов: атрибут **LANG** определен 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:

```
AddDefaultCharset koi8-r
AddLanguage ru .html .txt
```

2. Для версий Apache, начиная с 1.3.10:

```
AddCharset koi8-r .html .txt
AddLanguage ru .html .txt
```

3. Для более старых версий Apache (или если версия вам неизвестна):

```
AddType "text/html; charset=koi8-r" .html
AddType "text/plain; charset=koi8-r" .txt
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 придется 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 все готово. При просмотре ваших страничек никто теперь не увидит мусора из умляутов вместо нормальных русских букв, и не придется больше выбирать кодировки!

8.4.3 Проверка правильности кириллизации

Как проверить, удалось ли проставить 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`
(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`

8.4.4 Полезные советы

А совсем старые версии браузеров не показывают русских букв при таком способе, что делать?

Посоветовать пользователю сменить версию браузера на более новую. В качестве прецедента, 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 предпочитаю (3) чтобы пользователю было удобно и (2) чтобы новые версии работали (при этом стандартам соответствовало), а вы?

9 Текстовые процессоры

9.1 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 15.3 (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, 9.1.2 (teTeX) версии старше чем 1.0) включают в себя русификацию **out of box**. Признаком русифицированности LaTeX'a out-of-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 9.1.1 (T2)).

9.1.1 Пакет T2

Пакет T2 был изначально создан Werner Lembergom 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 приведен в разделе 9.1.2 (Русификация teTeX).

Пакет 'T2' можно найти на любом сайте CTAN в директории `macros/latex/contrib/supported/t2`, последняя версия лежит по адресу `ftp.vsu.ru` <<ftp://ftp.vsu.ru/pub/tex/t2-current.tar.gz>>

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, ko18-r, cp1251, cp866, iso8859-5)

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

Шрифты lh начиная с версии 3.20 (они наверняка есть в вашем дистрибутиве) поддерживают все нужные кодировки – смотрите раздел 15.3.1 (lh).

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` <<ftp://ftp.vsu.ru/pub/tex/ruhyphen.tar.gz>>

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 включен 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 отпадет.
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 особенностью T2-кириллизации является то, что "прогонке- tex-файла через latex вывод предупреждений и ошибок ведется на внутренней кодировки – то есть T2:

Артем 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>

9.1.2 Русификация teTeX

Описание русификации *teTeX* взято из FAQ, поддерживаемого для эхо-конференции **RU.TEX** Алексеем Махоткиным.

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
      =russian
      =english
```

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

```
texconfig init
```

Описанное выше можно проделать через стандартную конфигурационную программу `texconfig` – выбрать меню `/hyphenation`, далее `latex` и раскомментировать `russian`.

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 [9.1.1](#) (Пакет T2).

P.S. Это только один из вариантов, самый простой и универсальный, решения проблемы русификации. 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 `cyramstx`.

Чтобы добавить формат `cyrtext` (русский TeX), проще в каталоге `web2c` выполнить:

```
initex '\input cyrtext.ini \dump'
```

и бросить символическую ссылку с именем `cyrtext` 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.

9.1.3 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 будет такой роскоши, как автоматическая расстановка дефисов, но все 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 yu ya
```

Чрезвычайно сложно преобразовывать ваши русские тексты в такую кодировку, но вы можете автоматизировать этот процесс. Программа translit (раздел 12.2 (Символьная перекодировка)) поддерживает опцию вывода TeX.

9.2 StarOffice

Наиболее полную и подробную информацию по русификации StarOffice можно найти на страничке Леона Кантера по адресу <http://www.blackcatlinux.com/StarOffice/>

Последняя новость: Sun, который на корню скупил Star Division Corp., выпустил бесплатную версию русифицированную StarOffice 5.2, правда, шрифты для него нужны в ISO 8859-5 кодировке.

Off-topic: Следующая версия StarOffice (6.0?) выйдет под лицензией GNU – так было обещано (Sun слов на ветер не бросает).

9.2.1 StarOffice 5.2

Эта инструкция представляет собой дополнение к *Инструкции по инсталляции StarOffice 5.2* <<http://www.sun.com/products/staroffice/5.2/docs/RUSSIANSSETUP.PDF>>, отражающее особенности установки и настройки русской версии этого пакета. Вы узнаете, как наиболее полно реализовать возможности, заложенные в этом пакете, и обойти возможные проблемы.

Наиболее полную и подробную информацию по русификации StarOffice 5.2 для *Black Cat Linux 6.2* <<http://www.blackcatlinux.com/>> можно найти на страничке Леона Кантера по адресу <http://www.blackcatlinux.com/StarOffice/>

Шрифты и кодовые страницы *StarOffice 5.2* – не только первая русифицированная версия этого пакета, но и первая версия, которая может работать с различными кодировками кириллицы, то есть отображать установленные в системе шрифты в соответствии с текущей кодировкой, в которой запущен *StarOffice*. Таких кодировок поддерживается три – *KOI8-R*, *ISO8859-5* и *CP1251*. Текущая кодировка определяется пакетом по значению переменной окружения *LANG* – оно может иметь значения *ru_RU.KOI8-R*, *ru_RU.ISO8859-5* или *ru_RU.CP1251* (Для выяснения подробностей смотрите раздел 11.1.1 (Настройка локали)).

Внимание! *uk_UA.KOI8-U*, *uk_UA.CP1251*, *ru_UA.KOI8-U*, *ru_UA.CP1251* не поддерживаются! При запуске *StarOffice* с одним из этих значений *LANG* все надписи будут отображаться вопросительными знаками! Для работы с документами на украинском языке необходимо использовать *ru_RU.CP1251*.

StarOffice способен отображать установленные в системе шрифты в соответствии с текущей кодировкой, в которой запущен пакет. То есть, если в системе установлены шрифты *KOI8-R*, а текущая кодовая страница *StarOffice* – *CP1251*, он попытается "собрать" *CP1251* из *KOI8-R*, но при этом будут большие потери, так как в *KOI8-R* отсутствуют не только украинские, белорусские и южнославянские буквы, но и многие типографские знаки, такие как кавычки-"елочки", длинное тире и другие. При импорте документов из *Microsoft Office* эти недостающие символы будут заменяться вопросительными знаками, что затруднит работу с импортируемыми документами. Поэтому мы рекомендуем для работы со *StarOffice* использовать шрифты в кодировке *CP1251*, которая содержит самый полный набор символов, из которых легко "собирается" как *KOI8-R*, так и *ISO8859-5*. *StarOffice* использует два вида шрифтов – растровые (pcf) для элементов интерфейса (меню, подсказки, формы и др.), и TrueType – для печати и отображения документов на экране. Чтобы повысить качество отображения документов на экране, можно использовать шрифты TrueType из Windows. Далее подробно рассказывается, где взять и как правильно установить все необходимые шрифты.

Установка шрифтов Информацию о том, где можно найти растровые шрифты для X Window, и о том, как их установить, следует поискать в разделе 15.2 (Шрифты X window) и/или 4.1 (Шрифты для X Window). Вам нужно достать и установить шрифты в кодировке CP1251.

Информацию об установке TrueType шрифтов можно найти в разделе 15.4 (Шрифты TrueType).

Информацию об установке TrueType шрифтов можно найти в разделе 15.1 (Шрифты TrueType).

Установка StarOffice Перед установкой убедитесь, что в вашем домашнем каталоге доступно не менее 260Мб. Процесс установки подробно описан в *Инструкции по установке StarOffice 5.2* <<http://www.sun.com/products/staroffice/5.2/docs/RUSSIANSSETUP.PDF>>.

Подключение шрифтов TrueType для печати Для того, чтобы работать с кириллицей в *StarOffice*, необходимо сначала добавить новые шрифты к драйверу печати *StarOffice* – библиотеке *XPrinter*. Для этого в пакет входит специальная утилита – *SPAdmin*. Выполните ее командой `~/office52/program/spadmin` (~ означает домашний каталог). После этого нажать кнопки "Add Fonts" -> "Browse" и указать каталог, где лежат TrueType шрифты – например, `/usr/share/fonts/MICROSOFT-CP1251/Type1`.

Далее подтвердите установку всех перечисленных шрифтов. Подсистема печати *StarOffice* готова к работе.

Выбор кодировки для работы со StarOffice Как уже упоминалось выше, *StarOffice 5.2* может работать в одной из трех кодировок: *KOI8-R*, *ISO8859-5* и *CP1251*. Каждая из этих кодировок имеет свои достоинства и недостатки. Попробуем разобраться.

1. *KOI8-R*. Единственное достоинство – позволяет отправлять письма и сообщения в группы новостей в этой же кодировке, и в ней же сохранять документы .html. Все остальное – сплошные недостатки: слишком много вопросительных знаков при импорте из *Microsoft Word*.

2. *ISO8859-5*. Единственная кодировка, которая позволяет работать со встроенными в эту версию словарями для проверки русской орфографии и переносов. Имеет также большинство украинских (кроме "г с чубом") и белорусские буквы. Недостатки: письма, отправленные в этой кодировке, не читаются большинством доступных клиентов, в том числе *netscape messenger*'ом. Кроме того, при нажатии кнопки "отображать непечатаемые символы" вместо привычных значков наблюдаются русские буквы "З- и "Ж".
3. *CP1251*. Наиболее удачная, на наш взгляд, кодировка для тех, у кого нет проблем с русской орфографией: 100% импорт из *Microsoft Office*, полный набор специальных символов, включая Евро, возможность работать на любом из славянских языков. Недостатки – русский словарь недоступен, письма отправляются в *ISO8859-5*.

Вывод: если нужна почта и новости – работаем в *KOI8-R*, нужен словарь – *ISO8859-5*, во всех остальных случаях – *CP1251*.

Для запуска *StarOffice* в кодировке, отличной от системной, можно указать полное имя 11.1.1 (locale) непосредственно в командной строке, например:

```
LANG=ru_RU.CP1251 ~/office52/program/soffice
```

Либо же можно подредактировать сам файл `~/office52/program/soffice`, или файлы *kdlnk*, или написать свои сценарии для запуска *StarOffice* в различных кодировках.

Для ввода украинских букв достаточно сменить раскладку клавиатуры – например, при помощи переключателя *kkb*.

Печать из StarOffice Для вывода на принтер *StarOffice* генерирует данные в формате *PostScript*. Дополнительная русификация интерпретатора *Ghostsript* не требуется, так как все необходимые шрифты встраиваются в документ.

Как это происходит? Попробуем разобраться, для чего производились все перечисленные выше действия. Как уже упоминалось, в качестве драйвера печати *StarOffice* использует библиотеку *Bristol Xprinter*. Эта библиотека может использовать шрифты только в формате *Type1*, если шрифт содержит более 256 символов – используются только первые 256, в порядке, описанном в векторе кодировки шрифта. В комплекте со шрифтом обязательно должен быть файл метрики в формате *.afm*.

При установке шрифтов программой *SPAdmin* информация о шрифте добавляется в файл *xp3/psstd.fonts* в формате, сходном с форматом файлов *fonts.dir/fonts.scale*. В каталоге *xp3/pssoftfonts* и *xp3/fontmetrics/afm* создаются символические ссылки соответственно на файлы шрифта и метрики. После этого *Xprinter* "узнает" о том, что у него появился новый шрифт.

При запуске *StarWriter* запрашивает у *Xprinter* список установленных шрифтов. Только шрифты, перечисленные в *psstd.fonts*, появятся в списке доступных шрифтов *StarWriter*'а. А для отображения на экране из X-сервера запрашивается первый попавшийся шрифт, у которого в названии второе поле (Family) совпадает с тем, что написано в *psstd.fonts*, при этом поля Foundry и Encoding игнорируются. То есть, если в *Xprinter* установлен шрифт в одной кодировке, а в X – в другой, использовать их в *StarOffice* практически невозможно. Это происходит со стандартными шрифтами *Times*, *Helvetica* и *Courier*.

Проблемы при работе со StarOffice Основная проблема заключается в том, что в этой версии безнадежно испорчены все фильтры для работы с форматами, которые не предусматривают указания кодировки – *text*, *rtf*, *Word6.0/95*. При сохранении в любом из этих форматов русские буквы заменяются на "?". То есть обмен документами с пользователями Windows возможен только через формат *Word/Excel 97/2000*. Выше упоминалось, что некоторые почтовые клиенты плохо переносят письма в кодировке *ISO8859-5*, несмотря на явное указание этой кодировки. Кроме того, мы не рекомендуем использовать функцию замены стандартных шрифтов документа (Сервис - Параметры - Текстовый документ - Стандартные шрифты), так как это может вызвать проблемы.

10 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**, который 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 "С", 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 создаете открытку для вашей подруги. В этом случае вы, вероятно, будете нуждаться в печати документа с различными шрифтами и т.д. И это уже определенно 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 программы, 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.

10.1 Преобразование текста в PostScript

Иногда у вас есть простой ASCII KOI-8 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 "матерым", легко настраиваемым программным продуктом с большим количеством опций и настроек. Она позволяет управлять форматами, размещениями страниц, выделением и т.д. Другая утилита, делающая примерно то же самое (теперь доступная как часть проекта GNU) – *enscript*.

10.1.1 A2ps конвертер

Преобразователь текста в PostScript был и остается 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>
```

10.1.2 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 своем составе, но их очень просто доустановить. Как это сделать, описано ниже (спасибо Michael Van Canneyt):

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* <<ftp://ftp.redhat.com/pub/contrib/i386/enscript-fonts-koi8-1.0-1.i386.rpm>>.
3. Если вы не используете RPMки, вытяните файл `textbook.tar.gz` из архива на *sunsite.unc.edu* <<ftp://sunsite.unc.edu/pub/academic/russian-studies/Software/>> (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:

```
mkafmmap *.afm
```

4. Установка завершена. Попробуйте напечатать текст в KOI8-R следующей командой:

```
enscript --font=Textbook8 --encoding=koi8 some.file
```

10.1.3 rtxt2ps

Если же вам требуется действительно быстрый и простой способ, а качества вывода для вас не критично, и все, что вам нужно – это только русский текст на бумаге, попробуйте пакет *rtxt2ps* <<http://www.siber.com/sib/russify/converters/>>. Это очень простой, без украшательств конвертер текста в PostScript. Качество вывода не очень хорошее (честно говоря, плохое), но это работает.

10.2 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 12.2 (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 9.1.3 (Работа с Washington Cyrillic)).

10.3 Кириллица в PostScript

Эксперты говорят, что PostScript – это просто. Я не могу судить – у меня было слишком много вещей, которые надо изучить, чтобы выкроить время для изучения PostScript. Но я все равно попробую использовать мои небольшие знания об этом вопросе. **Я буду очень благодарен за любую информацию об этом вопросе от вас, друзья, которые знают больше меня** (вас приблизительно 99% от Земной популяции).

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

- шрифт кириллицы *загружен* в принтер, и тогда у вас все 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=pwrite'`.

Кроме описанного метода, можно включить шрифты в файл Postscript руками с помощью обычного текстового редактора – подробно это описано в разделе 10.4 (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**).

10.4 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 создается 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 LaTeXом. 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 `tlascii` (см. `tlutils`).

Для вставки вместо редактора можно использовать скрипт `includeresources` из `psutils` – он делает то же, но автоматически.

Этот же прием срабатывает и для обычного Postscript файла.

10.5 Ghostscript

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

10.5.1 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 заместить 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.

10.5.2 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 [15.1](#) (Шрифты `Type1`).

10.6 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 *Internet* (например, посмотрите на *архив SimTel* <<ftp://ftp.simtel.net>>).
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 **Alt** шрифты. Для этого имеются две причины:

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

Следует отметить, что в Болгарии вместо "альтернативной кодировки" используется кодировка *МИК*.

Однако, в этом случае вы должны будете преобразовать ваши тексты из **KOI8-R** в **Alt** перед ссылкой их на принтер. Это не проблема, так как имеется множество программ, делающих это (для примера смотрите раздел [12.2](#) (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)`.

10.7 Формат А4

Это имеет весьма отдаленное отношение к кириллизации, но иногда очень полезно знать, что надо сделать, чтобы ваш текст был отформатирован под страницу формата А4 (заимствовано из Serbian-HOWTO).

- *dvips* – измените нужным образом файл `/usr/share/texmf/dvips/config/config.ps` или `~/.dvips`.
- *ghostscript* – при вызове программы из командной строке, добавьте параметр `-sPAPERSIZE=a4`.
- *ghostview* – добавьте следующую строку в файл `~/.Xresources`:

```
Ghostview.pageMedia:  A4
```

- *xdvi* – добавьте следующую строку в файл `~/.Xresources`:

```
XDvi.paper:  A4
```

11 Локализация и Интернационализация

Пока я описывал, как заставить различные программы понять кириллицу. Обычно каждая программа требовала, чтобы это был ее собственный метод, как правило, чрезвычайно отличный от других. Кроме того, у некоторых программ была незавершенная поддержка языков, отличных от английского, не говоря уж об их неспособности взаимодействовать, используя родной язык пользователя вместо английского.

Проблемы, перечисленные выше, сильно подавляют, так как программное обеспечение редко создается только для местного рынка. Переработка существенных частей программного обеспечения каждый раз при входе на новый международному рынок очень неэффективна; а интернациональная поддержка, осуществляемая собственными средствами программы, уникальным и присущим только ей способом, в терминах долгосрочного планирования также не блестящая идея.

Следовательно, возникает потребность в стандартизации. И стандарт есть.

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 Соединенных 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 еще в процессе, но много ее 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.

11.1 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 Великобритании / 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 страничке *Локализация, как она есть* <www.sensi.org/~alec/locale/index.html>.

11.1.1 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 иксовой локали 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 путем 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 выдает, 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 путем 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) (2.2 (Portable Character Set)).

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 ее 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"
```

Согласно стандарту допустимы также короткие именованя 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 2.2 (Символы и кодировки)). Лучше не пользоваться значениями по умолчанию, а указывать точное длинное имя. Во 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, Codeset ISO8859-5 по умолчанию).

Некоторые могут пожелать сделать себе локализацию в другом наборе символов: ru_RU.X-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 map-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"
```

Не рекомендуется использовать строку окружения:

```
$ export LC_ALL={язык}
```

поскольку формально такой категории локализации нет, она "виртуальная" и обозначает "одно- временно 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. Включается она заданием строки окружения LANG="ru_RU.UTF-8" для России.

Устаревшая процедура установки Locale В RedHat Linux (как, вероятно, и во многих других дистрибутивах Linux), имеются фактически две *базы* данных locale: одна для библиотеки C (libc), а другая - для X библиотек. В идеальном случае должна иметься только одна база данных locale для всего.

Чтобы изменить значение locale по умолчанию, обычно достаточно установить системную переменную LANG. Например, как это делается в sh:

```
LANG=ru_SU
export LANG
```

Вы можете проверить действие этой команды сразу же, если запустите команду date. Результатом должен быть вывод дня, недели и месяца на русском языке.

RedHat 5.x определяет KOI8-R locale как ru_SU. Более очевидное название ru_RU используется для locale, основанного на iso-8859-5 кодировки.

Иногда вы можете захотеть изменить только один аспект locale без изменения других. Например, вы можете захотеть (Бог знает почему) пользоваться ru_SU locale, но печатаемые числа должны будут соответствовать стандарту POSIX один. В подобных случаях имеется набор системных переменных, которые Вы можете задать, чтобы сконфигурировать соответствующие части locale. Например, в нашем случае это бы выглядело так:

```
LANG=ru_SU
LC_NUMERIC=POSIX
export LANG LC_NUMERIC
```

Подробнее см. locale(7).

Теперь давайте держаться поближе к специфике Linux. К сожалению, в Linux libc версии 5.3.12 (пример: дистрибутив RedHat 4.1), отсутствует русская locale. В данном случае ее надо скачать из Interneta (я, однако, не знаю точного адреса).

Чтобы проверить, для каких языков у вас есть locale, выполните 'locale -a'. Это выведет список всех locale из баз данных, доступных libc.

Что касается библиотек X, то они имеют свою собственную базу данных locale. В версии XFree86 3.3 уже имеется российская база данных locale. Я не уверен, есть ли она в предыдущей версии. В любом случае вы можете проверить это, изучив директорию /usr/lib/X11/locale/ (в большинстве систем). В моем случае уже есть подкаталоги, именованные koi8-r и даже iso8859-5.

11.1.2 Locale зависимое программирование

С locale программа не должна знать о различных символьных преобразованиях и правилах сравнения, описанных выше. Вместо этого они используют специальный API, который действует по правилам, определенным locale. Кроме того, нет необходимости для программы пользоваться только одной locale для соблюдения всех правил – возможно пользоваться другими правилами, описанных 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, воз-
вращенные 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:

```
if (c >= 'A' && c <= 'Z') {
    ...
```

Подобные конструкции с точки зрения правильного "localeпрограммирования" 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 удаленного файла в MS-DOS кодом 0x0E5 – не очень хорошее решение. Еще примеры плохих решений: знаменитая русская буква "Н- в редакторе GoldEd или русская буква "р- в Norton Commander...

2. 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.

3. 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 11.2 ().
4. EUC/Unicode поддержка. На этом уровне, программа не использует тип `char`. Взамен это она использует `wchar_t`, который определяет объекты, достаточно большие, чтобы содержать символы Unicode. ANSI C определяет этот тип данных и соответствующий API.

Для выяснения подробностей смотрите, например, (7 ()) или (5 ()).

11.2 интернационализация

В то время как локализация описывает, как адаптировать программу к иностранному окружению, **интернационализация** (или **i18n** для краткости) детализирует способы общения программы с неанглоговорящим 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. Я не предполагаю описывать методы программирования еще и потому, что `gettext` пакет поставляется с превосходным руководством.

Просьба о сотрудничестве: Если вы хотите изучить пакет `gettext` и сделать свой вклад в проект GNU или просто сделать вклад без всякого изучения, то вы можете сделать это! GNU становится международным, так что все утилиты делаются locale зависимыми. Проблема состоит в том, чтобы переводить сообщения с Английского языка на Русский (и другие языки, конечно если захотите). В общем, что следует сделать: вы должны получить специальный .po файл, состоящий из Английских сообщений для неких утилит, и связать каждое сообщение с его эквивалентом на русском. В конечном счете, это заставит говорить систему по-русски, если пользователь захочет этого! Для для подробностей войдите в контакт с Ulrich Drepper (drepper@ipd.info.uni-karlsruhe.de).

12 Совместимость

Следовать стандарту – это еще не все. В реальной жизни надо обеспечить еще 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 **Alt (cp866)** или в **cp1251**. Также должна быть возможность запускать русскоязычные программы из MS-DOS.

В большинстве случаев (кроме HTTP) достаточно обеспечить преобразование данных в **KO18-R**. Если мы говорим о данных с простой структурной организацией, то это просто – смотрите раздел 12.2 (Символьная перекодировка).

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

12.1 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 (MPЕG файлы), моя программа приема электронных сообщений автоматически декодирует его и запустит МРЕG проигрыватель.

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 *KOИ8*.

Ниже перечислены MIME-законы, которые описывают правила работы с обычными текстами и текстами в `richtext` формате, использующих другие кодировки, отличные от *KOИ8*. Вы можете вставить эти директивы в один из файлов конфигурации 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 перекодировки смотрите раздел 12.2 (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
```

Достаточно, это работает только в случае обычного текста. Бинарные файлы данных должны сами обрабатывать 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. До недавнего времени Microsoft Mail использовал испорченный механизм работы с MIME. Хотя и Netscape Navigator/Communicator клиент отправки/приема сообщений известен тем, что шлет текст письма в **cp1251**, а его заголовок в `charset=koi8-r` кодировке и наоборот.

Довольно часто случается так, что те, кто посылают вам письма, неправильно настраивают (или не настраивают вообще, а по умолчанию стоит черте-че – 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 `~/.prosmailrc` со следующим содержанием (но я *предупреждаю* вас, что после ошибки в `~/.prosmailrc` 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"
```

12.2 Символьная перекодировка

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

С моей точки зрения, утилиты должны быть стандартными. В нашем, частном, случае такой "стандартной" утилитой является *GNU recode*. (<http://www.iro.umontreal.ca/pinard/recode/>).

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

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

Так же широко распространены *iconv* (входит в поставку *glibc*)

Опять же cp1251 в koi8-r:

```
$ iconv -f866 -tKOI8-R -o<outfile> infile
```

Можно воспользоваться пакетом *translit*. Он поддерживает много популярных кодировок и даже способен создавать TeX файлы (см. раздел 9.1 ()) на русском языке.

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

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

Для других программ преобразования – загляните на страничку *SovInformBureau* или на *ftp.funet.fi*. Вы даже можете использовать специальную моду для *emacs* (Смотрите раздел 6.2 (*Emacs* и *XEmacs*)).

12.3 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 дает 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*:

```
codepage=866,iocharset=koi8-r
```

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

```
/dev/hda5 /mnt/d vfat noexec,dev,suid,rw,umask=002, codepage=866,iocharset=koi8-r, gid
```

для выяснения подробностей смотрите *man mount*.

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

12.4 Поддержка кириллицы в DOS эмуляторе

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

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 r драйвер Вади-ма Курлянда (последняя версия этого драйвера – 2.0, и его развитие прекращено). Он прекрасно настраивается и поддерживает много кодировок, в том числе Alt и KOI8. Однако он, вероятно, не будет работать для X Window (по крайней мере, версия 1.14 не работает).

Оба русификатора можно найти в большинстве Российских Internet архивов, например, в *ftp архиве Курчатовского Института Ядерной Физики* <<ftp://ftp.kiae.su/pub/cyrillic/msdos>>.

Кроме того, есть патч в <http://www.ice.ru/vitus/misc/>, который позволяет обойтись без русификатора клавиатуры. Этот патч рекомендуется в X-ax, так как позволяет избежать двух клавиш переключения: одной в DOS, другой – на остальном десктопе. Для работы с удаленным 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](ftp://ftp.dosemu.org) <ftp://ftp.dosemu.org/dosemu/Development/vga_cyr8x16.pcf.gz>, или взять у автора (эти шрифты сделал Дима Болховитянов) www.inp.nsk.su/bolkhov <<http://www.inp.nsk.su/~bolkhov/files/fonts/vga4dosemu/>>.

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_ruscii_8x16.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 4.1 (Настройка шрифтов 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 А. Страхова, и *cyrkeyb* Pete Kvitck; однако лучше воспользоваться патчем <http://www.ice.ru/vitus/misc/>.

12.5 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-r):

```
style "gtk-default-ru" {
    fontset = "-cronyx-helvetica-medium-r-normal--*-**-*-*--koi8-r,\
-cronyx-courier-medium-r-normal--*-**-*-*--koi8-r,\
-cronyx-fixed-medium-r-normal--*-**-*-*--koi8-r,\
-cronyx-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 <http://www.gtk.org/tutorial/gtk_tut-21.html>, а также <http://www.linux.org.ru/gnome/>

13 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.

Для настройки кириллицы прежде всего полезно прочитать разделы 3 (Настройка текстовой моды), 4 (Настройка X Window), и только после этого, зная, что вы делаете, обратиться к подразделам этого параграфа.

13.1 Suse

13.1.1 Настройка текстового режима

Пример русификации консоли koi8. Проверено для версии 6.3

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*" выберете пункт "*Change configuration file*". Среди предложенных для изменения переменных нас интересуют:

`CONSOLE_FONT` – консольный шрифт (как поменять эту переменную, было описано ранее; можно менять ее и здесь),

`CONSOLE_SCREENMAP` – установите переменную равной `koi2alt`,

`CONSOLE_MAGIC` – установить равной "К- (кавычки не писать),

`KEYTABLE` – рекомендую установить равной `ru3.map.gz` (переключатель правый *Alt*, 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, отредактировав файл `rc.local` в `/etc/`.

P.S. Если при установке указать, что ваш язык – "русский- (`LANGUAGE = russian`)", то все настроится автоматически (ну, почти).

13.1.2 X window

Делайте так же, как описано в разделе 4 (X Window) - настройка ничем от стандартной настройки не отличается.

13.2 Linux RedHat 5.2

13.2.1 Настройка текстовой моды

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*.

13.2.2 Клавиатура

Клавиатура в *RedHat 5.2* на нижнем уровне управляется 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/qwerty/*, 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*

13.2.3 Консоль

Консоль в *RedHat 5.2* на нижнем уровне управляется 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 посылать *ESC* (К. Можно посылать его, например, из `~.profile`)

13.2.4 X window

Делайте так же, как описано в разделе 4 (X Window) – настройка ничем от стандартной настройки не отличается.

13.3 Debian

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

13.3.1 Настройка русской локали

В файле `/etc/environment` исправьте `LANG=C` на `LANG=ru_RU.KOI8-R`. Если вы хотите, чтобы у `root` были русские сообщения и он мог вводить русские буквы в X Window, сделайте тоже самое в `/root/.bashrc`.

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`, скажите вы. Но `ru_RU` уже существует, это локаль с кодировкой *ISO-8859-5*. Поэтому следует поставить `ru_RU.KOI8-R`.

13.3.2 Русификация текстового режима.

Если в системе все еще 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.map.gz /etc/console-tools/default.map.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 'Ю'. Чтобы побороть это, можно сделать свой тип терминала (в *debian* есть *linux-koi8*). Но гораздо проще и грамотнее поставить `sr866` шрифты и включить таблицу перекодировки. Последний шаг – копирование карты клавиатуры. *ru1* рассчитана на 102-кнопочную клавиатуру без *Microsoft*'овских клавиш.

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

Вставьте в `/etc/kbd/config`:

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

Выполните команду

```
# cp /usr/share/keymaps/i386/qwerty/ru1.map.gz /etc/default.map.gz
```

с привилегиями пользователя root.

13.3.3 Русификация X Window System

Установите пакет *xfonts-cyrillic* со второго CD. Затем выполните команду

```
# ln /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:*. Иначе, 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.

13.4 Slackware 7

Этот раздел является сильно сокращенной версией (только кириллизация) *Slackware-7-INSTALL-RUS.HOWTO* (c) 1999 Вячеслава Калошина (multik@sw.ru, multik@istu.edu)

13.4.1 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$n
done
loadkeys ru.map
```

таким образом, появятся русские буквы в консоли. Шрифты переключаются правым Ctrl.

- Положить файл *.inputrc* следующего содержания в каталог */etc/skel:*
-

```

$if Bash
    set convert-meta off
    set output-meta on
    set input-meta on
    set show-all-if-ambiguous on
    "\e[1~" beginning-of-line
    "\e[2~" " "
    "\e[3~" delete-char
    "\e[4~" end-of-line
    "\e[C" forward-char
    "\e[D" backward-char
    "\e[A" previous-history
    "\e[B" next-history
    "\e[21~": "exit\C-m"          # F10 - Exit from shell
$endif

```

здесь на *F10* "повешана" команда `exit`.

- 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 (ее можно взять в других дистрибутивах, или скачать и развернуть *отсюда* <<http://linux.irk.ru/doc/linuxrus/locale.tar.gz>>) в `/usr/share/locale/ru_RU/`.
- от `root`'а 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.

13.4.2 Русификация X Window System

Делайте так же, как описано в разделе 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` <<http://xtalk.price.ru/linux/.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 вышеприведенный способ русификации не будет работать.

14 Кириллический Офис

14.1 Текстовые процессоры Раздел 9 (Текстовые процессоры) вынесен за рамки раздела 14 (Кириллический Офис) в силу своей чрезвычайной важности.

14.2 Графические редакторы

Здесь описывается настройка графических программ для визуализации и вывода на печать русских букв.

14.2.1 tgif

Ответ дан Александром Киселевым в эхоконференции `fido7.ru.tex`.

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-шрифты, 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 идее, включает использование локализованных `.po` файлов. Ну да ладно – главное, что работает.

14.2.2 xfig

В `fido7.ru.linux` Борис Тоботрас кидал два, как он выразился, "гнусных хака- для русификации `xfig` – они лежат где-нибудь, или надо самому выкладывать?

Тот же Борис в `fido7.ru.tex` дал следующий совет по использованию `xfig` при работе с `TeX/LaTeX`.

Если вы хотите использовать картинки, сделанные в `xfig` при работе с `TeX/LaTeX`, по Вам поможет пакет `psfrag`. В `xfige` пишете вместо надписей тэги – типа `Petr`, `moment` и т.д. Экспортируете картинку в `eps`. А в `TeX`овском 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. Преимущество 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 постскрипте, а в pdf, то приходится извращаться: создавать одностраничные документы в TeXе (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.

15 Кириллические шрифты

Здесь будут ссылки на кириллические шрифты – как на платные, так и на свободные наборы. Александр Бердников из `fido7.ru.tex`, цитата о том, можно ли найти Postscript шрифты для набора научных текстов:

"На настоящий момент практически единственный продавец на рынке шрифтов – это ParaType. Остальные (в том числе и "свободные"- сайты типа Vedi или Litera) предлагают в основном акцидентные шрифты (качество не проверял), 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 (то есть, какие реально у тебя будут права, какие реально есть гарантии и какую реально поддержку и сопровождение обеспечивает Paratype). Кроме того, необходимо озаботиться 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.

15.1 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 <ftp://ftp.kapella.gpi.ru/pub/cyrillic/psfonts/>](ftp://ftp.kapella.gpi.ru/pub/cyrillic/psfonts/). 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 шрифтов – например, Pushkin.

В разделе 10 (Вывод на печать и PostScript) можно найти дополнительную информацию по использованию Type1 шрифтов.

15.1.1 literat

В 1996 году компания ParaGraph International и Russian Foundation for Basic Research сделал шрифты literat доступными для бесплатного использования (подробнее смотрите файл license из самого пакета). По отзывам, начертание этих шрифтов сильно отличается (причем далеко не в лучшую сторону) от того, что изначально называлось "гарнитура литературная- (старые математические книжки 60-х годов) – шрифт явно не доделан (можно посмотреть и ужаснуться, как выглядит слово "Где"), но в 99% результат вполне удовлетворительный.

Сами шрифты доступны по адресу [ftp.vsu.ru <ftp://ftp.vsu.ru/pub/tex/literat.zip>](ftp://ftp.vsu.ru/pub/tex/literat.zip)

Для того, чтобы эти шрифты можно было использовать с LaTeXом, следует достать так же и пакет *literat-*.tar.gz* <<ftp://ftp.vsu.ru/pub/tex/>> – "виртуальные шрифты- для LaTeX (T2A, OT1, T1 и TS1 кодировки). Подробности по установке смотрите в файле описания этого пакета.

15.1.2 PSCYR

Набор шрифтов Type1 для работы с TeX/LaTeX – не совсем ясно как там с лицензией – по сути является результатом перевода шрифтов из формата ttf в Type1 (сам эти шрифты не смотрел, но по слухам качество не очень, поэтому никто из праводержателей на эти гарнитуры и не возмущается). Пакет состоит из двух частей:

1. PsCyr-type1-*.tar.gz – содержит шрифты Type 1 и их метрики. В пакет включены гарнитуры: Academy, Advertisement, Antiqua, College, Handbook, Lazurski, Magazine, Textbook, ArialCyrMT, CourierCyrPS, ERKurierKOI8RRegular, TimesNRCyrMT.
2. PsCyr-tex-*.tar.gz – содержит информацию о установке, TeX метрики, описание шрифтов, различные кодировки и примеры использования.

Найти можно по адресу [ftp.vsu.ru <ftp://ftp.vsu.ru/pub/tex/>](ftp://ftp.vsu.ru/pub/tex/)

Автор пакета: *Konstantin Chumachenko*.

15.1.3 Paradissa

Басиль К. Малышев в 1993 году выложил набор кириллических шрифтов при условии отсутствия модификации пакета (подробности смотри в лицензии пакета). Сам пакет можно найти в любом архиве CTAN в директории `fonts/cm/ps-type1/paradissa`. Этот пакет имеет некоторые технические "опечатки", которые можно исправить с помощью простого скрипта *'corr-cmc.sh'*, приведенного ниже. Скрипт использует утилиты `tldisasm` и `tlasm` из пакета `tlutils`.

```
#!/bin/sh

for i in cmc*.pfb; do

tldisasm $i | awk '
    /\^\/Subrs [0-9]* array$/ {s=1}
    /\^readonly def$/ && s {print "|-";s=0;next}
    /\^\/.notdef {$/ {while ($0 !~ /} |-$/) {getline}
    printf "/.notdef {\n\t0 0 hsbw\n\tendchar\n"}
    {print}
' | tlasm -b > .temp

rm -f $i
mv -f .temp `echo $i | sed 's,\(cmc.....\)\.pfb,\1.pfb,` ``

done
```

15.1.4 LucidaSO

Благодаря свободно распространяемому дистрибутиву StarOffice 5.2 (См. раздел 9.2.1 (StarOffice 5.2)) 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/> <[ftp.vsu.ru](ftp://ftp.vsu.ru)>. Там лежат и сами шрифты, и все, что надо для их использования с LaTeX и PDFLaTeX.

В пакет входят 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 9.1.2 (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 регенерации `map` файлов;
- для того, чтобы посмотреть 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@vsv.vsu.ru.

15.1.5 dmtr-fonts

Это полностью свободный и сделанный с нуля шрифт Дмитрия Сорокина. Найти можно по адресу <ftp://ftp.chg.ru/pub/Linux/mandrake/7.0RE/disk1/Mandrake/RPMS/dmtr-fonts-0.99-ipl1mdk.noarch.rpm> <[dmtr-fonts-0.99-ipl1mdk.noarch.rpm](ftp://ftp.chg.ru/pub/Linux/mandrake/7.0RE/disk1/Mandrake/RPMS/dmtr-fonts-0.99-ipl1mdk.noarch.rpm)>

15.1.6 sharatype

Шрифты Сергея Шарашкина, которые можно найти по адресу [sharatype-1.0b-ipl1mdk.noarch.rpm](ftp://ftp.chg.ru/pub/Linux/mandrake/7.0RE/disk1/Mandrake/RPMS/sharatype-1.0b-ipl1mdk.noarch.rpm) <<ftp://ftp.chg.ru/pub/Linux/mandrake/7.0RE/disk1/Mandrake/RPMS/sharatype-1.0b-ipl1mdk.noarch.rpm>>, из которых два – свободные, а два оказались сомнительными, так как сделаны на базе шрифтов Peter Soos.

15.1.7 Всякая всячина

www.d-s.ru

www.slavonic.orthodoxy.ru – церковно-славянские шрифты.

15.2 Шрифты для X-window

Наиболее часто встречающиеся растровые шрифты перечислены ниже.

Кроме растровых шрифтов, в X Window можно использовать Type1 Postscript шрифты, правда, их не стоит использовать как терминальные (обычно в этих шрифтах отсутствуют необходимые для этого символы), но в некоторых случаях они предпочтительнее, так как являются масштабируемыми – смотрите раздел 15.1 (Шрифты Type1).

15.2.1 Cronux

В X Window, еще в конце 1995 года, в качестве стандартного пакета был включен набор **KOI8-R** шрифтов, разработанных *Cronux*. 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, включенного 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. Например, у меня были проблемы с *Marple V* для Linux, который падал по непонятным причинам с пакетом *vakufonts*, но прекрасно работал со "стандартными" набором.

Шрифт *-cronyx-fixed-** с полным соответствием *koi8-r*, в том числе с рамками: <http://www.ice.ru/lists/cyrfonts>.

Коллекция *Cronux* имеет новые версии, а также переработанный (и по-моему лучший) вариант *by Serge Vaculenko*, который включает все славянские кириллические буквы, а также и некоторые "дореволюционные" русские буквы.

15.2.2 Cyr-RFX

Кроме распространенных шрифтов *Cronux*, существуют гораздо более эргономичные 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.

15.2.3 xrus

Пакет *xrus* (не путать это с программой *xruskb*, ранее известной как *xrus*) является модификацией изначальных шрифтов *Cronux*. Его можно найти по адресу <ftp://ftp.kiae.su/cyrillic/x11/fonts/xrus-2.1.1-src.tgz>.

Последнюю версию пакета *xrus* можно найти на страницах Андрея Чернова: <http://koi8.pp.ru/>. Теперь их адрес <http://koi8.pp.ru/dist/xbrus-2.3.1-bin.tgz> <[x6rus-2.3.1-bin.tgz](http://koi8.pp.ru/dist/xbrus-2.3.1-bin.tgz)> (и версия – 2.3.1). Возможно, Чернов будет менять их место, так что когда дело дойдет до *beta*, можно спросить в списке *cyrfonts* о предпочтительных адресах.

15.2.4 European

Очень благородные "изовые" шрифты можно взять на любом зеркале GNU в каталоге *intlfonts*, например, *European.tar.gz* (131Кб) <<ftp://ftp.chg.ru/pub/gnu/intlfonts/intlfonts-1.2-split/European.tar.gz>>. В Debian они уже есть.

15.2.5 xsyr

Пакет *xsyr* можно найти здесь: <http://sawsoft.newmail.ru/LS/koi8-extended.html> <sawsoft.newmail.ru/LS/koi8-extended.html>.

15.3 Шрифты для TeX/LaTeX

При работе с TeX/LaTeX можно использовать Type1 шрифты – где их можно найти, описано в разделе 15.1 (Шрифты Type1).

Внимание: При использовании Type1 шрифтов в тексте, насыщенном математическими символами, восприятие текста умеренно ухудшится (если не принимать дополнительных усилий по согласованию шрифтов). Математические CM-шрифты не будут сочетаться с текстовыми шрифтами.

Ниже описаны стандартные кириллические шрифты *LaTeX*, которых хватает в 99.99% случаев.

15.3.1 lh

Лучшие кириллические шрифты всех времен 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 (что-то говорили про Concreate – Конкретная Математика). Шрифты *lh* всегда можно достать по следующим адресам:

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

Начиная с версии 3.20 (?) *LH* поддерживает стандартные кодировки *LaTeX* 9.1.1 (T2A, T2B, T2C, X2), кроме того, поддерживаются LCY, OT2 и др.

15.4 Шрифты TrueType

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

Несколько шрифтов в формате Unicode сама Microsoft выложила для бесплатного пользования (я только не уверен в лицензии – можно ли их использовать вне Windows) – так что идите на страничку *Microsoft Typography* <<http://www.microsoft.com/typography/fontpack/default.htm>> и ищите, пока не найдете.

Благодаря своей популярности шрифтов в этом формате (в том числе и кириллических) в сети можно найти в очень большом количестве; только, как правило, они имеют не очень высокое качество, и пока мало программ под Unix/Linux могут использовать их без "напильника и кувалды". Но времена меняются – поддержка ttf уже встроена в X-window, с помощью gnu-библиотеки *freetype* некоторые программы умеют выводить шрифты на экран и в печать (например, текстовый редактор *yudit*).

Если кто знает, где можно найти качественные и бесплатные (и не очень) ttf шрифты – то пусть сообщит мне.

16 Библиография

1. Андрей Чернов. *KOI-8* <<http://www.nagual.pp.ru/~ache/koi8.html>>. KOI-8: информация и настройка.
2. Ulrich Drepper. *Интернационализация в проекте GNU* <<http://i44www.info.uni-karlsruhe.de/~drepper/conf96/paper.html>>. Полное описание подхода GNU к делу i18n.
3. Michael Karl Gschwind. *Интернационализация* <<http://www.vlsivie.tuwien.ac.at/mike/i18n.html>>. Различные ссылки по теме i18n.
4. Сергей Наумов. *Информация о кириллическом программном обеспечении* <<http://sunsite.oit.unc.edu/sergei/Software/Software.html>>. Информация о настройке кириллицы.
5. The Open Group *Single UNIX specification* <<http://www.UNIX-systems.org/online.html>>.
6. RFC 1489 *RFC 1489* <<file://ds.internic.net/rfc/rfc1489.txt>>
7. Alec Voropay. *Интернационализация. Что это?* <<http://www.sensi.org/~alec/locale>>. Использование locale на русском.

17 Полезные ссылки

17.1 Авторская страничка

Авторская домашняя страничка <<http://www.inp.nsk.su/~baldin>>

17.2 Общая информация

17.2.1 Linux

Общая информация по Linux <<http://www.linux.org>>

17.2.2 XFree86

XFree86 FTP site <<http://www.xfree86.org>>

17.2.3 Локализация

Локализация, как она есть <<http://www.sensi.org/~alec/locale/index.html>> – страничка о локализации (locale), и обо всем, что с этим связано.

17.2.4 Рекомендации IANA

Рекомендации IANA <<http://www.isi.edu/in-notes/iana/assignments/character-sets>>

17.2.5 "Группы новостей"

Группа новостей *relcom.fido.ru.linux*.

Группа новостей *relcom.fido.ru.unix*.

17.3 Различные полезные программы

17.3.1 Postscript

Ghostscript <<http://www.cs.wisc.edu/~ghost/index.html>>

Страничка *a2ps* <<http://www-inf.enst.fr/~demaille/a2ps.html>>

GNU *enscript* <<ftp://ftp.gnu.org/pub/gnu>>

17.3.2 Словари

Русский словарь для GNU *ispell* <<http://www.ispras.ru/~knizhnik>>

17.3.3 TeX/LaTeX

Программа-фильтр для T2 Latex пакета <<http://www.inp.nsk.su/~baldin/misc/atex.c>>

Пакет *translit* <<ftp://192.148.249.121/pub/central\protect\T2A\textunderscoreeastern\protect\T2A\textunderscoreeurope/russian/translit/translit.tar.Z>>. Если у вас проблемы с обратными слэшами – просто удалите их.

17.3.4 X Window

Программы по работе с BDF-шрифтами Антона Зиновьева <<ftp://lml.bas.bg/home/anton/linux/trscripts-1.2.tar.gz>>

Пакет *kbd* для Linux <<ftp://sunsite.unc.edu/pub/Linux/system/keyboards/>>

Пакет *xruskb* <<ftp://ftp.relcom.ru/pub/x11/cyrillic/>>

17.3.5 Текстовые редакторы

Пакет *russian.el* для *emacs* <<http://www.math.uga.edu/~valery/russian.el>>

17.4 Шрифты

17.4.1 X-window

Шрифты Дмитрия Болховитянова из проекта *Cyr-RFX* <<http://www.inp.nsk.su/~bolkhov/files/fonts/cyr-rfx/>> - очень рекомендую.

Cronyx <<http://www.cronyx.ru>> – создатели кириллических шрифтов для X Window.

Русские шрифты для X <<ftp://ftp.kiae.su/pub/cyrillic/x11/fonts/xrus-2.1.1-src.tgz>>

Русские X шрифты в cp1251 (Windows) кодировке <<http://www.kiarchive.ru/pub/cyrillic/x11/fonts/winfonts4X.tar.gz>>

17.4.2 Type1

Русские шрифты для *Ghostscript* и *StarOffice* <<ftp://ftp.kapella.gpi.ru/pub/cyrillic/psfonts>>

17.4.3 TrueType

Microsoft Typography <<http://www.microsoft.com/typography/fontpack/default.htm>>

17.5 Различные кириллические ресурсы

Cyrillic resources at KIAE <<ftp://ftp.kiae.su/pub/cyrillic/>>

Cyrillic resources at RELCOM <<ftp://ftp.relcom.ru/pub/cyrillic/>>

17.5.1 Русские ссылки

linux.org.ru Общая информация по Linux на русском <<http://www.linux.org.ru>>

linux.ru.net Общая информация по Linux на русском <<http://www.linux.ru.net>>

Cyrillic resources at FUNET <<ftp://ftp.funet.fi/pub/culture/russian/comp/>>

Полезные кириллические пакеты <<ftp://sunsite.unc.edu/pub/academic/russian-studies/Software>>

Collection of Cyrillic resources <<ftp://ftp.ccl.net/pub/central\protect\T2A\textunderscoreeastern\protect\T2A\textunderscoreeurope/russian>>. Если у вас проблемы с обратными слэшами – просто удалите их.

17.5.2 Белорусские ссылки

Все о белорусизации Линукса. <<http://bellinux.sourceforge.net>>

17.5.3 Болгарские ссылки

Linux Society in Bulgaria. <<http://linux.home.bg/>> Эти парни полагают большие усилия, чтобы "болгаризация- Линукса не делалась стихийно и необдуманно. Здесь вы найдете много полезных ссылок и информацию. Если вы желаете начать какой-нибудь проект, то подпишитесь пожалуйста на список рассылки *develop@linux.home.bg* или *dict@linux.home.bg*, в зависимости от характера вашей работы.

Здесь довольно успешно пытаются рассказать обо всем. <<http://linux.gyuvet.ch/>>

Информации, новости и гр. <<http://linux-bg.exco.net/>>

ВРАТАТА <<http://www.openbg.net/thegate/>> – новости, ссылки электронного журнала "Phreedom".

SAdict <<http://sa.dir.bg/linux/>> – Англо-болгарский и болгаро-английский электронный словарь.

Пакет для болгаризации Линукса. <<http://lml.bas.bg/~anton/linux/bglinux.html>>

17.5.4 Македонские ссылки

Все для македонской кириллицы. <<http://agitator.excelland.com/>>

Новости, македонская локализация KDE <<http://slavej.pmf.ukim.edu.mk>>

17.5.5 Сербские ссылки

Сербская кириллица в Интернете <<http://www.srpsko-pismo.org/>>

17.5.6 Украинские ссылки

The Linux Ukrainian HOWTO <<http://www.ci.net.ua/NOC/ukr/>>

Советы для использования КОИ8-У. <<http://www.terena.nl/projects/multiling/koi8-u/index.html>>

17.5.7 cyr-asian

По адресу *ftp.logic.ru* <<ftp://ftp.logic.ru/pub/logic/linux/cyr-asian/>> 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).