Проблема множественности русских кодировок и новое семейство кириллических шрифтов *

Л.Н. Знаменская, С.В. Знаменский

25 июня 2000 г

Существование множества широко распространенных таблиц кодировок документов на русском языке под DOS, UNIX и другими операционными системами создает проблемы при работе, например, в сети, так как ASCII файл, отправленный из другой системы, становится совершенно непригодным для чтения. Новое семейство шрифтов, стандарт TDS размещения файлов в системе и другие средства позволят пользователям TEX писать по-русски, не задумываясь о кодировках

TEX стал одним из наиболее известных средств общения между учеными различных регионов. Для того, чтобы решить проблему несовместимости различных русификаций TEXа, Российский Фонд Фундаментальных Исследований (РФФИ) выдвинул идею создания некоммерческого дистрибутива русского TEXа. В 1996 году началась работа при поддержке РФФИ над проектом "Русский TEX". Основная задача проекта — сделать естественным обмен информации клиентов сети, работающих на разных платформах и с разными операционными системами.

Новый стандарт TDS (TEX Directory Structure) является совершенным фундаментом для создания такой системы. Проблема, которая здесь возникает, специфична для языков, основанных на кириллице, и она связана с множественностью кодировок кириллических шрифтов. Например, существует несколько широко используемых кодировок русских шрифтов под UNIX. Даже Microsoft® использует совершенно разные кодировочные таблицы русских текстов под DOS и Windows® для одного и того же ПК. Соответственно, можно найти в различных директориях CTAN METAFONT исходники кириллических шрифтов имеющих одно и тоже имя, например, cmrz10, и в то же время имеющих различные коды для русской буквы "А".

*Эта работа выполнена при поддержке Российского фонда фундаментальных исследований, грант № 95-07-19400в

1 Описание проблемы

Распространяющееся во многих регионах России программное обеспечение основано на принципах, предполагающих создание специальных шрифтов для каждой группы языков народов России. Такой подход принуждает народности регионов к созданию своих шрифтов. В результате, подготовленные к публикации материалы оказываются непригодными для редактирования в другом месте, цитирования в многоязычном документе. Это создает значительные трудности не только профессиональным лингвистам, но и библиотекарям и другим специалистам, связанным с материалами, издаваемыми на различных языках. Если сейчас не создать единого способа представления многоязычной информации, то в недалеком будущем развитие Internet может привести к тому, что культуры многих даже не очень малых народностей регионов России окажутся вне рамок объединяющего информационного пространства общей мировой культуры. Перед нами стоит задача обеспечить разработчиков программных продуктов научно-обоснованным способом представления лингвистической информации, а филологов, работников библиотек и издательств - инструментом для создания статей и книг, файлы которых могут читаться, редактироваться и распечатываться с высоким качеством в любой точке России.

2 Шрифты

Первое, что необходимо было сделать — это выбрать подходящее кириллическое расширение стандартного для TEXa семейства CM (Computer Modern) шрифтов и подобрать новые имена таким образом, чтобы в них нашла отражение и кодовая таблица. Так как LH шрифты CyrTUG нельзя было использовать бесплатно для некоммерческого дистрибутива РФФИ, то мы обратились к Н. Глонти и А. Самарину за разрешением использовать их шрифты, так как они являются первыми и наиболее широко используемыми TEXовскими шрифтами в России. Через полтора месяца мы получили любезное согласие авторов шрифтов использовать эти шрифты или модифицировать шрифты или их исходники для дистрибутива РФФИ и мы благодарны авторам за столь великодушное решение. К сожалению, мы не могли ждать столь долго и именно в то время, когда было получено разрешение от Н. Глонти и А. Самарина, создание нового русского расширения семейства CM шрифтов находилось на завершающей стадии — на стадии создания кернинга.

При создании нового семейства шрифтов мы преследовали следующие цели:

- Сохранить оригинальные исходники CM шрифтов без изменения для того, чтобы обеспечить идентичность при печатании латинских текстов новыми шрифтами
- Сделать текст и буквы более привычными русскому глазу, сохраняя традиционные для CM шрифтов особенности

- Обеспечить более равномерную затемненность текста
- Сделать все исходники CM шрифтов, включая гарнитуру concrete, доступными для печати на русском языке
- Исключить, по возможности, ошибки при автоматической генерации шрифтов для устройств с низким разрешением
- Заложить фундамент для будущей поддержки всех алфавитов народов России, базирующихся на кириллице

Мы использовали CM макросы, фрагменты исходных CM текстов и лишь в незначительной степени фрагменты CMCYR исходных текстов. Однако мы сочли полезным сохранить порядок контрольных точек.

Когда создание нового семейства шрифтов близилось к завершению, было решено сравнить качество печати новыми шрифтами и наиболее широко распространенными шрифтами Самарина и Глонти. Большой математический текст был распечатан на 10 и 12 точек на 600-точечном HP LaserJet4 принтере с помощью этих семейств шрифтов. К этим текстам был приложен лист чистой бумаги для того, чтобы эксперты могли сравнить и высказать свое мнение о качестве шрифтов. Эксперты РФФИ (физики и математики) сравнивая шрифты, отметили, что оба семейства русских шрифтов достаточно качественные.

А как быть с именами новых шрифтов? Первая идея — использовать стандартную схему имен шрифтов. Но если мы пойдем по этому пути, то получим, что имя расширенного 8-битного шрифта слишком сильно отличается от имени соответствующего стандартного 7-битного CM шрифта. Как следствие этого, у пользователя возникнут проблемы с адаптацией новых стилей и использованием примитивных TEXовских команд выбора шрифтов. Для того, чтобы избежать таких проблем, мы решили начинать имя нового шрифта с букв RF (Russian Font + Russian Foundation), третий символ в имени (цифра) используется для обозначения кодовой таблицы и конец имени — такая же последовательность символов как в конце имени соответствующего CM шрифта.

3 Проблема множественности русских кодировок

Мы должны разрешить возникающую в типичных ситуациях проблему: когда система TEXовских файлов расположена на сервере, а клиенты работают в разных операционных системах с различными кодировками. Главная проблема — выбор подходящей процедуры для ввода TEXовского файла в любой кодировке.

Наше решение проблемы заключается в создании такой программы, которая не только корректно определяла бы кодировку кириллического файла, но и автоматически осуществляла бы перекодировку его в соответствии с локальной кодировкой.

Почему бы и нет? Каждый, кто знает русский язык, легко отличит текст в правильной кодировке, от того же самого текста в другой кодировке. Но если мы внимательно посмотрим на проблему, то увидим множество препятствий на этом пути.

3.1 Взаимно однозначное соответствие кодовых таблиц как часть проблемы

Если двоичный файл был случайно определен как кириллический, то мы должны иметь возможность восстановить ошибочно конвертированный файл. Работа же в сети имеет гораздо больше трудностей, так как при неоднократном перекодировании мы должны сохранить первоначальную информацию. Мы не видим иного пути решения этой проблемы кроме как иметь в своем распоряжении фиксированный алгоритм перекодировок. Было бы естественно сохранить первые 128 ASCII позиций кодовой таблицы без изменений, а между оставшимися 128 в каждой паре кодовых таблиц установить взаимно однозначные соответствия, причем эти соответствия для каждой пары таблиц должны быть согласованы.

К несчастью это невозможно, поскольку множество символов в этих частях таблиц сильно разнятся при переходе от одной кодовой таблицы к другой. Такие образом, мы должны разрешить символу менять свое значение при перекодировке, в то же время мы должны постараться уменьшить множество таких возможных изменений значения. Подходящим решением этой проблемы является разбиение множества всех возможных символьных значений на 128 классов эквивалентности таким образом, что любая перекодировка меняет значение символа только внутри его класса эквивалентности. Но это также невозможно, так как некоторые значения неизбежно окажутся в разных классах эквивалентности, и лучшее, что мы можем сделать в такой ситуации — это использовать наименее употребимые значения.

3.2 Другая проблема кириллических языков

Существует более 60 языков, основанных на кириллице, и некоторые из них не имеют установленных кодовых таблиц. Многие файлы содержат в себе нетекстовые команды в силу того, что большинство программных продуктов, как правило, вставляет в файл служебные символы и поэтому программа должна по-возможности правильно отличать кириллическое слово от такой последовательности символов.

Мы не можем использовать только информацию о множестве символов данного файла для того, чтобы правильно определить кодовую таблицу документа. Другой причиной на наш взгляд является то, что разные кодовые таблицы используют одно и тоже множество символов. Подсчет того сколько раз каждая буква появляется в тексте может оказаться недостаточным для того, чтобы правильно определить кодировку короткого файла. Существует более точный инструмент: подсчитать число возникновения в документе каждого двухбуквенного сочетания.

Этот эффективный подход требует более чем 128 K памяти для хранения промежуточной информации. Алгоритм, выполняющий собственно статистический анализ этих данных, включает в себя умножение вычисленных логарифмов и недостаточно быстр особенно на ПК. Каким путем следует пойти для того, чтобы получить приемлемый результат просто и быстро?

Идея состоит в выборе двух множеств из множества всевозможных двухбуквенных сочетаний: множества A — множества часто встречающихся в кириллических текстах двухбуквенных сочетаний и множества U — обычно не используемых таких сочетаний. Далее надо посчитать числа N_A и N_U таких сочетаний из множеств A и U соответственно, появившихся в файле. Число $C = \frac{N_A - N_U}{N_A + N_U}$ и покажет нам выглядит ли этот файл как кириллический текст или нет. Такое значение должно быть вычислено для каждой известной кодовой таблицы, большее из этих значений и определит правильно кодовую таблицу файла. Это кажется нам наиболее быстрым, легким и эффективным способом правильного определения кодировки файла, поскольку наиболее часто используемые сочетания двух символов (менее 5% всех сочетаний) дают более 50% всех двухбуквенных сочетаний в русском тексте и примерно половина всех возможных сочетаний практически не используется в русском языке. Остается только проблема собственно выбора множеств A и U.

3.3 Как мы выбирали A и U

Огромную помощь нам оказала уникальная книга Гиляровского и Гривнина [9] с примерами текстов на практически всех языках. Мы ввели эти примеры в компьютер и посчитали число появления всевозможных двухбуквенных сочетаний. Но здесь возникла новая проблема: как быть с “не русскими” буквами?

Как уже отмечалось, у большинства языков, основанных на кириллице, не существуют фиксированные кодовые таблицы. Нам также не известны какие-либо попытки использования русской клавиатуры и специальных TEX команд для печати на большинстве кириллических языков России, Монголии и Аляски. Поэтому для каждого языка, в алфавите которого присутствуют нерусские буквы, мы создавали два файла: в первом файле использовали символы для представления нерусских букв в соответствии с таблицами, приведенными выше, а во втором файле для представления таких букв использовали последовательность символов, начинающуюся с символа “/” (например, /ЪК для “К с клювом”, или /КЦ для “К с полочкой”, или /ЛЬ для Љ, или /Ц для Ц) и использовали максимально стандартные TEX последовательности для акцентов. Для русского языка мы использовали три разных текста и словарь, содержащий 51924 слова. Остальные языки были представлены единичными файлами. В нашем распоряжении имелось 109 файлов для 64 языков.

Конечно же мы не могли быть уверены в том, что кто-то другой будет использовать те же самые коды или последовательности символов для обозначения нерусских букв, поэтому при подсчете двухбуквенных сочетаний в каждом файле мы выделили в отдельную группу все

буквы с неизвестными кодами, все ASCII символы, не являющиеся буквами, выделили также в отдельную группу и отделили все латинские буквы, не используемые в кириллических текстах. После подсчета мы определили двухбуквенные сочетания, которые никогда не появляются в кириллических текстах. Таких сочетаний оказалось 695, они и образовали множество U.

Выбор множества A был немного сложнее. После нескольких попыток осуществить этот выбор, мы пришли к следующему алгоритму. Для каждой пары букв и каждого файла вычислялся логарифм относительной частоты появления. Для того, чтобы избежать бесконечности, мы заменили нулевые частоты очень маленькими ненулевыми значениями как если бы эти двухбуквенные сочетания появились бы в файле вдвое длиннее. Затем мы вычислили суммы по всем файлам и использовали их для выбора. Выделилось 314 часто встречающихся пар, состоящих только из русских букв, для которых почти каждое слово содержит такие двухбуквенные сочетания. Нам требовалось обойти эффекты возможного использования других TEXовских ключевых слов для нерусских букв или местной кодовой таблицы, которая может соответствовать лишь русской части одной из наших кодовых таблиц. Поэтому мы использовали только 306 пар, исключив двухбуквенные сочетания, которые могли бы быть порождены нашими конкретными обозначениями нерусских букв.

Так был завершен алгоритм распознавания кодовой таблицы

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

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

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

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

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

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

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

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

[9] Р.С. Гиляревский, В.С. Гривнин. Определитель языков мира по письменностям. М.: Nauka, 1964.