ЭЛЕКТРОННЫЕ ПУБЛИКАЦИИ В СРЕДЕ INTERNET И МНОЖЕСТВЕННОСТЬ КОДИРОВОК РУССКОГО ЯЗЫКА*

А. В. Дорофеев
Новосибирский государственный университет, Россия
А. М. Федотов
Институт вычислительных технологий СО РАН
Новосибирск, Россия
e-mail: fedotov@adm.ict.nsc.ru

The paper analyzes the problems caused by the existence of many various systems of coding Russian files in electronic publications and the methods of their solution for WWW electronic publications are described employing an identifying and re-coding proxi-server implemented on the SB RAS server.

Сеть Internet является самой большой в мире компьютерной сетью, объединяющей в единое информационное пространство миллионы компьютеров во всем мире. Число пользователей этой сети удваивается в среднем каждые полгода. Колоссальными темпами растет количество информационных ресурсов, доступных через сеть Internet. Однако существует целый ряд проблем, препятствующих развитию сети, одной из которых является множественность кодировок национальных алфавитов, в частности русского языка. Например, только кодировок арабского алфавита существует более 30, а кириллического алфавита около 10 (широко используются только 5). Вопрос унификации кодировок неоднократно поднимался на различных конференциях, но пока еще далек от какого-либо решения. Русскоязычные серверы работают во всех широко используемых кодировках, и пользователям сети Internet приходится приспосабливаться к существующему положению. Если опытный пользователь, в принципе, всегда может определить, в какой кодировке работает сервер, и настроить свое программное обеспечение (просмотрщик — браузер) на кодировку сервера, то работа поисковых "роботов", которые ориентируются на смысловое содержание текста, наталкивается пока на непреодолимые трудности, связанные с распознаванием текста.

В настоящей статье предлагаются и обсуждаются способы решения проблемы кодировок для публикаций русскоязычных текстов в среде Internet (на WWW или при помощи SQL-баз данных).

*Работа выполнена при финансовой поддержке Российского фонда фундаментальных исследований, грант № 97-07-90372.

© А. В. Дорофеев, А. М. Федотов, 1997.

1. Национальные кодировки

Начиная с середины 80-х гг. компьютерная подготовка документов, применяемая как для создания твердых копий и оригиналов-макетов печатных изданий, так и для использования в электронном виде (передача по каналам связи или в различных информационных системах), все больше и больше вытесняет обычные способы — машинопись или типографский набор. В большинстве организаций уже давно исчезли печатные машинки и наборные автоматы — их заменили персональные компьютеры.

Постоянно совершенствуется и технология подготовки оригиналов-макетов научно-технических текстов. Такие текстовые процессоры, как TEX, MS Word, PageMaker и другие, им подобные, позволяют подготовить для печати практически любой документ. Однако, если речь идет о текстах, содержащих математические символы, то пока непревзойденной является издательская система TEX, созданная Д. Кнутом.

Вместе с компьютерным набором возникли две проблемы, связанные с использованием символов, не содержащихся в стандартной кодовой таблице (Code Page) символов us-ascii (ISO-ASCII): первая — это проблема включения в подготавливаемые тексты символов национальных алфавитов и акцентированных символов, даже в странах, использующих в качестве основного латинский алфавит; вторая — это проблема использования специальных символов, какими, например, набираются математические формулы.

Рассмотрим проблему представления символов национальных алфавитов. Первый способ решения этой проблемы (а также и проблемы, связанной с представлением математических символов) заключается в следующем. Имея оригинал-макет, подготовленный для печати, и используя гипертекстовой протокол передачи данных (HTTP — HyperText Transfer Protocol), вы можете представить пользователю по сети графический образ страницы. Но такой способ передачи публикаций приемлем только для небольших объемов информации (например, аннотации статей из журнала) и требует от пользователя необходимости работы только в графическом режиме и с быстрыми каналами связи, что для нашей страны вряд ли можно считать приемлемым.

Другой способ решения проблемы представления символов, отсутствующих в стандартной кодовой таблице, заключается в передаче пользователю-клиенту не самих документов, а их образов, подготовленных для печати и просмотра в специальных форматах, таких как PostScript или Adobe PDF. Этот способ уже можно назвать “удовлетворительным”, хотя и он не свободен от ряда недостатков.

Во-первых, для документов, подготовленных в этих форматах, резко возрастает объем передаваемой информации по сравнению с их текстовым аналогом, хотя он и меньше, чем соответствующий объем графических файлов.

Во-вторых, для просмотра документов требуется установка на компьютер пользователя специального математического обеспечения (свободно распространяемого, например Ghostscript или AdobeAcrobat), что у многих пользователей иногда вызывает противодействие.

В-третьих, применение специальных форматов затрудняет совместную работу над документами и их дальнейшее использование.

Наконец, в-четвертых, это пока еще достаточно неудобно, так как, прежде чем посмотреть документ, вам необходимо перенести его на свой компьютер, а потом загрузить специальную программу для его просмотра.

В принципе, такой способ можно считать приемлемым для документов, подготовленных и хранимых в системе MS Word, если клиент использует для просмотра WWW-

страниц просмотрщик фирмы Microsoft, хотя сам формат MS Word не решает полностью проблему совместимости не только для текстов с символами кириллицы, но и для текстов, содержащих символы только латинского алфавита. Однако следует отметить, что объем документа в формате MS Word значительно превышает объем документа в формате PostScript. Этот недостаток, связанный с загрузкой файлов, подготовленных в формате PostScript или PDF, по-видимому, явление временное, так как уже разработаны просмотрщики WWW-страниц, интегрированные с просмотрщиками PostScript файлов.

Для европейских языков, алфавиты которых созданы на основе латинского, проблема включения в электронные документы символов национальных алфавитов более или менее удовлетворительно решилась с появлением системы UNICODE, использующей расширенную кодовую таблицу ASCII (256 символов). В этом случае символы национальных алфавитов включаются во вторую половину расширенной кодовой таблицы (числовые коды в диапазоне от 128 до 255), однако для всех европейских языков одного расширения (128 дополнительных символов) кодовой таблицы не хватило.

В качестве стандарта, принятого Международной организацией стандартизации (ISO, International Standards Organisation), используется 10 различных расширений кодовой таблицы (Code Page):

Кодовая страница ISO 8859-1 (старое название Latin 1) поддерживает языки Западной и Центральной Европы: албанский, немецкий, английский, каталонский, датский, испанский, финский, французский, фламандский, ирландский, исландский, итальянский, голландский, норвежский, португальский.

Кодовая страница ISO 8859-2 (Latin 2) поддерживает славянские языки Центральной Европы: хорватский, венгерский, польский, румынский, словацкий, словенский, чешский, а также немецкий.

Кодовая страница ISO 8859-3 (Latin 3) поддерживает языки: эсперанто, галицийский, мальтийский, турецкий.

Кодовая страница ISO 8859-4 (Latin 4) поддерживает языки Восточной Европы: эстонский, латышский, литовский.

Кодовая страница ISO 8859-5 поддерживает кириллический алфавит, языки: болгарский, белорусский, македонский, сербский, русский, украинский. Эта кодировка в настоящий момент принята к использованию государственным стандартом Российской Федерации (ГОСТ).

Кодовая страница ISO 8859-6 поддерживает арабский алфавит (в расширенной системе передачи данных пока не используется).

Кодовая страница ISO 8859-7 поддерживает греческий алфавит.

Кодовая страница ISO 8859-8 поддерживает арабский алфавит (в расширенной системе передачи данных пока не используется).

Кодовая страница ISO 8859-9 (Latin 5) расширение таблицы Latin 1, связанное с дополнительными буквами исландского (кельтского) языка.

Кодовая страница ISO 8859-10 (Latin 6) другой более полный вариант кодовой таблицы для языков Восточной Европы, включая скандинавские языки.

В настоящее время ведется разработка стандартов кодовых таблиц, включающих символы всех языков мира, в том числе китайского и японского[1].

Введение различных кодовых таблиц и локализация языковых стандартов на компьютерах полностью сняли национальную проблему использования различных текстовых про-

[1]С точки зрения авторов, эта работа вряд ли будет когда-либо завершена, поскольку трудно найти дизайнера, который был бы в состоянии создать шрифт, содержащий около 16 тысяч символов

цессоров для подготовки оригиналов-макетов. Отметим, что стандартные операционные системы на компьютерах таких фирм-производителей как IBM или Sun Microsystems поддерживают приведенные выше национальные языковые стандарты.

Однако проблемы передачи электронных версий документов из страны в страну и представления специальных символов по-прежнему остались. При передаче документов через Internet проблема определения кодовой таблицы, в которой подготовлен соответствующий документ, решается путем использования расширения MIME (Multipurpose Internet Mail Extensions) и обязательного задания Content-Type для протокола HTTP с соответствующим именем кодовой таблицы (например, charset=ISO-8859-1). Если на вашем компьютере установлена операционная система, поддерживающая UNICODE (или MIME), то вы имеете возможность увидеть текст в том виде, в котором его подготовил автор. Эта возможность "теоретически" реализуется при правильной настройке вашего компьютера и просмотрщика WWW-страниц (браузера) и наличии соответствующих графических (или экранных для текстового просмотрщика) шрифтов. Практически это не всегда реализуется корректно, так как не все просмотрщики работают по стандарту. Вернее, авторам не известен ни один просмотрщик, который полностью удовлетворял бы требованиям стандарта.

2. Русские кодировки

С русскими кодировками дело обстоит сложнее, чем с европейскими. В силу сложившихся в нашей стране традиций, международный стандарт кириллического алфавита ISO-8859-5 (charset=ISO-8859-5) мало применяется. Эта кодовая страница (рис. 1) в основном используется только на компьютерах с ОС UNIX, в частности на компьютерах фирмы Sun Microsystems с операционной системой Solaris. С использованием данной кодировки выполнена локализация ряда программных продуктов для UNIX-совместимых систем, например нескольких крупных систем управления базами данных (в частности, Oracle).

Code	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
80 :																
90 :																
A0 :		Ё														
B0 :	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
C0 :	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я
D0 :	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
E0 :	р	с	т	у	ф	х	ц	ч	ш	щ	ъ	ы	ь	э	ю	я
F0 :		ё														

Рис. 1. Кодовая таблица ISO-8859-5.

Кроме кодировки ISO-8859-5 в настоящее время достаточно широко применяются четыре другие кодировки символов кириллицы (Code Page).

KOI8-R (charset=KOI8-R). Кодовая страница KOI8-R (рис. 2) базируется на устаревшем государственном стандарте кода обмена информацией (КОИ) и применяется в основном на компьютерах с ОС UNIX на базе платформы Intel. Данная кодировка нашла отражение в документе RFC1489 (Request For Comments). Де-факто, с легкой руки Relcom'а, эта кодировка признана как русский сетевой стандарт сети EuNet/Relcom.

Поскольку Relcom пока является монополистом в сфере доставки электронной почты, рекомендуется почтовые сообщения отправлять в кодировке KOI8-R, чтобы быть уверенным в том, что адресат вас поймет правильно.

Code	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
80 :																
90 :																
A0 :				ё												
B0 :				Ё												
C0 :	ю	а	б	ц	д	е	ф	г	х	и	й	к	л	м	н	о
D0 :	п	я	р	с	т	у	ж	в	ь	ы	з	ш	э	щ	ч	ъ
E0 :	Ю	А	Б	Ц	Д	Е	Ф	Г	Х	И	Й	К	Л	М	Н	О
F0 :	П	Я	Р	С	Т	У	Ж	В	Ь	Ы	З	Ш	Э	Щ	Ч	Ъ

Рис. 2. Кодовая таблица KOI8-R.

CP866 (charset=x-CP866). Кодовая страница фирмы Microsoft для IBM совместимых компьютеров (рис. 3) применяется в основном на персональных компьютерах с операционной системой MS DOS 6.22. Иногда ее неправильно называют “альтернативной” кодировкой. Альтернативная кодировка (CP855 — charset=ibm855 или charset=CP855) сейчас практически не применяется и отличается от кодировки CP866 расположением нескольких букв и некоторых знаков (при использовании кодировки CP866 рекомендуется исключать букву ё).

Code	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
80 :	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
90 :	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я
A0 :	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
B0 :																
C0 :																
D0 :																
E0 :	р	с	т	у	ф	х	ц	ч	ш	щ	ъ	ы	ь	э	ю	я
F0 :																

Рис. 3. Кодовая таблица CP866.

CP1251 (charset=Windows-1251 или charset=x-CP1251). Кодовая страница фирмы Microsoft (рис. 4), принятая в графических системах семейства MS Windows.

Code	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
80 :																
90 :																
A0 :									Ё							
B0 :									ё							
C0 :	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
D0 :	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я
E0 :	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
F0 :	р	с	т	у	ф	х	ц	ч	ш	щ	ъ	ы	ь	э	ю	я

Рис. 4. Кодовая таблица Windows-1251.

MACOS (charset=x-ru-MAC). Кодировка (рис. 5), используемая на компьютерах фирмы Apple Macintosh.

Code	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
80 :	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
90 :	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я
A0 :	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
B0 :																
C0 :																
D0 :																
E0 :	р	с	т	у	ф	х	ц	ч	ш	щ	ъ	ы	ь	э	ю	я
F0 :																

Рис. 5. Кодовая таблица x-ru-mac.

Кроме них существуют совсем уже забытые кодировки: основная (или болгарская) и ГОСТ.

Такое разнообразие кодировок связано с тем, что в нашей стране в основном используются “персональные компьютеры” Intel’овской архитектуры. Количество компьютеров с другой архитектурой составляет доли процента от их общего числа. И, как следствие этого, в области программного обеспечения мы находимся в условиях гигантской экспансии фирмы Microsoft и ее системных продуктов, которая пытается навязать всему миру свои стандарты. Поэтому в области Internet-технологий кодировка CP1251 имеет у нас в настоящее время наиболее широкое распространение. Другие кодировки из приведенного выше списка связаны либо со старыми программными средствами той же фирмы Microsoft, либо с устаревшим стандартом, применяемым в среде UNIX на Intel’овской платформе.

К сожалению, одними обвинениями в адрес фирмы Microsoft, что она не соблюдает международные и национальные (для национальных локализаций своих продуктов) стандарты, дело не решить. Конечно, до тех пор, пока операционные системы Microsoft работали только в локальных сетях организаций, особых проблем с совместимостью не возникало. Но после появления таких программных продуктов, как MS Windows-95 и MS Windows NT 4.0 компьютеры под их управлением вышли в глобальные сети и навязали пользователям свои стандарты, которые разработчики вынуждены учитывать при создании систем электронных публикаций. Это касается не только русского языка, но и языков стран Центральной и Восточной Европы.

Разработчики распространенных пользовательских оболочек для Internet, рассчитанных на среду MS Windows (Netscape Navigator, MS Internet Explorer, Eudora, Pine и др.), были вынуждены смириться с этим обстоятельством и снабдить свои программные продукты средствами “совместимости” с международными и национальными стандартами. Правда, эти средства не везде работают одинаково хорошо.

3. Способы решения проблемы совместимости

Ввиду того, что документы на русскоязычных серверах могут храниться в различных кодировках, а пользователь-клиент может понимать далеко не все кодировки, возникает задача корректной передачи электронных документов пользователю и отображения текста программой, с помощью которой пользователь просматривает электронный документ.

Наилучшим решением была бы реализация такого подхода, который предоставил бы пользователю возможность работы с электронными документами в языковой (кодовой) среде, используемой его операционной системой, автоматически производя перекодировку всех данных. Отметим, что задача перекодировки данных является не такой уж простой, как кажется на первый взгляд. Основные трудности порождаются наличием преобразований кодировок, не имеющих обратных преобразований. Поэтому при применении ошибочного преобразования часть информации может быть утеряна.

Проблема кириллических кодировок может решаться различными методами в зависимости от способов хранения документов на сервере, организации взаимодействия сервера с клиентской программой и от пользовательской оболочки, которая применяется для просмотра документов. Однако пока ни один из существующих методов не дает решения проблемы.

Рассмотрим различные способы представления русскоязычных текстов, которые используются в настоящий момент на российских серверах.

3.1. Единая кодировка

Наиболее простой для администратора сервера метод решения проблемы — предоставление доступа ко всем документам в какой-либо одной кодировке. Как правило, по этому пути идут последователи Relcom'a — приверженцы UNIX-систем, которые размещают файлы в кодировке KOI8-R.

Существует достаточно устойчивое мнение считать кодировку KOI8-R универсальной для внутреннего представления документов не только на серверах, но и для текстовых процессоров типа TEX. Это мнение обосновывается тем, что в UNIX-системах отсутствует механизм внутренней перекодировки типа TCP, а шрифты с другими кодировками пока еще не имеют достаточного распространения (к тому же мы пока не научили UNIX понимать кодировку Windows-1251). Не ясно только, почему выбирается кодировка KOI8-R, а не ISO-8859-5, которая является международным стандартом.

Если WWW-сервер работает только в одной кодировке, то пользователь должен сам позаботиться об установке необходимых шрифтов и о конфигурировании своей программы просмотра. До середины 1996 г. (т. е. до принятия стандарта HTML 3.2) пользователю это сделать в большинстве случаев было несложно. Графические просмотрщики, работающие в средах MS Windows и XWindows (например, программные оболочки Netscape Navigator, MS Internet Explorer или Mosaic), позволяют принудительно задать шрифт с нужной кодировкой для отображения текста документов, а соответствующие шрифты имеются для разных кодовых таблиц.

При стандартном просмотре документа через графический просмотрщик, как правило, применяются два семейства графических шрифтов, установленных в используемой графической оболочке: это *пропорциональный шрифт* (обычно Times Roman) и *равноширинный шрифт* (обычно Courier). Просмотрщики, поддерживающие стандарт HTML 3.2 (Netscape и MS Internet Explorer), могут использовать и другие шрифты, установленные в системе. В HTML 3.2 считается стандартным применение в качестве дополнительного пропорционального шрифта третьего семейства шрифтов, которое называется *шрифт без засечек* (семейство шрифтов типа Arial, Lucida, Sans, Helvetica). Однако для дополнительного шрифта пока не предусмотрена возможность устанавливать его имя принудительно. Здесь и возникает основная трудность, связанная с использованием одной кодировки. Если используемая для просмотра кодировка не является кодировкой операционной системы клиента, то он вынужден держать на своем компьютере два[2] или более семейства шрифтов одного наименования, присвоив второму семейству нестандартное имя. Поэтому вместо текста, который набран дополнительным шрифтом, пользователь получит бессмысленный набор символов (поскольку стандартное имя шрифта, как правило, соответствует кодировке компьютера пользователя-клиента).

Стандарт HTML 3.2 допускает использование любого семейства шрифтов, если вам известно его имя на пользовательской машине. В случае, если данное семейство отсутствует, происходит подстановка *пропорционального шрифта*.

С другой стороны, если сервер организован правильно (то есть с соответствующими указателями charset в поле Content-Type), то пользователь может не устанавливать дополнительные шрифты на свой компьютер, но, к сожалению, пока нет просмотрщика, который бы полностью и корректно проводил перекодировку для всех кодовых таблиц

[2]Большинство российских серверов предоставляют клиенту возможность работать в двух кодировках: KOI8-R и Windows-1251, а серверы, работающие в одной кодировке, как правило, используют кодовую таблицу KOI8-R.

русского языка. Netscape Navigator в среде MS Windows (кроме Windows NT) корректно работает пока только с кодировками CP1251 и ISO-8859-5, а для кодировки KOI8-R он требует установки дополнительных шрифтов. MS Internet Explorer пока не понимает кодировку ISO-8859-5, а для кодировки KOI8-R некорректно работает со специальными символами, такими, как, например, “тире” (встроенная система перекодировки правильно перекодирует только текст). Mosaic не имеет средств поддержки MIME.

Таким образом, принудительный выбор кодировки имеет большие неудобства. Во-первых, в настоящее время вы вынуждены отказаться от полного использования возможностей языка HTML 3.2 (дополнительных шрифтов, спецсимволов и др.). Во-вторых, “блуждая” по сети и переходя с сервера на сервер, вы будете вынуждены менять настройки своего просмотрщика, так как разные серверы работают с разными кодировками. В-третьих, теряется возможность копировать фрагменты текста документа в буфер обмена и переносить их в текстовые процессоры без дополнительной перекодировки.

Возможно, это проблема временная и скоро разработчики программного обеспечения устранят это недостатки. Хотя Microsoft пока не собирается поддерживать кодировку ISO-8859-5.

Самым неприятным моментом, связанным с этим подходом, является использование форм (<FORM>) и Java-приложений. В этих конструкциях, с одной стороны, как правило, используется системный шрифт операционной системы, который, очевидно, будет отображаться неправильно, а с другой стороны, клиент не имеет возможности отправить запрос на сервер (заполнить форму) на русском языке. Поэтому, если сервер использует для формирования своих документов SQL запросы к базам данных, такой подход нельзя считать приемлемым, даже в случае, когда разработчики просмотрщиков устранят существующие в них огрехи, связанные с перекодировкой.

3.2. Множественная кодировка

Несмотря на то, что современные просмотрщики WWW-страниц имеют возможность выбора кодовой страницы и уже в своем большинстве понимают MIME, абсолютно корректно они работают пока только в “родной” языковой среде операционной системы.

Поэтому в настоящее время единственно правильным решением проблемы, связанной с отображением русскоязычных текстов, является предоставление пользователю информации во всех кодировках русского языка (по крайней мере, в наиболее распространенных кодировках KOI8-R и CP1251). Отметим, что этот способ пока является наиболее распространенным и используется на большинстве российских серверов. Чаще всего пользователю предлагается выбор из двух кодировок: KOI8-R и CP1251 (MS Windows). Кодировки CP866 (MS DOS), ISO-8859-5 и MacOS на российских серверах представлены значительно реже (заметим, что встречаются серверы, предлагающие русские тексты с латинской транслитерацией русских букв).

При использовании нескольких кодировок русского языка право выбора кодировки предоставляется, как правило, пользователю, хотя встречаются и исключения. Чаще всего на начальной странице WWW-сервера предлагается выбрать кодировку, в которой пользователь-клиент предпочитает общаться с сервером. В этом случае на сервере должно быть организовано хранение копий документов в нескольких кодировках (что требует дополнительного дискового пространства) или динамическое перекодирование документов в реальном времени перед отправкой их пользователю.

В целом выбор кодировки пользователем из меню пока является наиболее приемлемым

решением, однако оно вынуждает пользователя считывать лишнюю страницу с сервера лишь для того, чтобы выбрать на ней нужную кодировку, что не очень удобно, особенно для медленных линий связи.

Используя возможности протокола HTTP, можно упростить задачу для пользователя, заставив сервер попытаться самостоятельно распознать кодировку программы-клиента, после чего перекодировать документы из локальной кодировки, принятой на сервере, в кодировку клиента или автоматически переключить пользователя на страницы с нужной кодировкой. Для этого в заголовке запроса перечисление допустимых форматов документов (Content- Type) должна указываться требуемая кодировка. Программа-клиент должна быть для этого специальным образом сконфигурирована. В этом случае пользователь может не задумываться о том, в какой кодировке хранится информация на сервере.

Встречаются также серверы (например, WWW-сервер Объединенного института геологии, геофизики и минералогии СО РАН), которые ведут базу данных пользователей и кодировок, с которой они работают. В этом случае при первом обращении к серверу вы выбираете кодировку, а затем ваш выбор заносится в базу данных, в которой по IP-адресу компьютера определяется используемая на нем кодировка. При последующих обращениях к этому серверу с того же компьютера используется полученная ранее информация. Правда, такой способ неудобен тем, что при организации перекрестных ссылок между различными серверами информация о кодировках теряется и на сервере, который не ведет базу данных клиентов, вследствие чего трудно организовать корректную ссылку на документы такого сервера.

4. Способы решения представления документов на сервере

Перечислим все наиболее распространенные способы представления документов на сервере с использованием множественности кодировок русского языка.

1. Хранение всей базы документов одновременно в нескольких кодировках в разных подкаталогах либо на разных виртуальных серверах.

2. Использование одной базы документов и динамическая перекодировка в зависимости от подкаталога или адреса виртуального сервера.

3. Определение кодировки клиента по ключевым словам в имени просмотрщика (браузера).

4. Установка сервера на разных портах для разных кодировок над одной (с перекодировкой) базой документов.

5. Установка сервера (или нескольких серверов) на разных портах для разных кодировок над несколькими базами документов, соответствующих разным кодировкам.

6. Хранение базы адресов клиентов и их кодировок.

7. Обработка всех выдаваемых документов с помощью специальной программы на сервере (например, CGI-script).

Независимо от используемых способов предоставления клиенту документов реально используются только два метода: либо хранение документов во всех кодировках, либо динамическая перекодировка. У всех перечисленных выше способов решения проблемы кодировок есть свои достоинства и недостатки, однако ни один из них (кроме пятого — поддержки нескольких серверов, каждый из которых работает в своей кодировке) не обеспечивает полного и корректного решения проблемы. Главным их недостатком является

невозможность организации диалога с клиентом (заполнение форм, формирование запросов) в разных кодировках, во всех этих случаях общение с сервером возможно только в одной фиксированной кодировке.

При использовании первого способа или при организации нескольких серверов для каждой из кодировок приходится держать несколько копий документов, что неудобно при частом изменении базы документов. Способы второй и седьмой нарушают структуру базы данных обращения к серверу, тем самым не дают правильно формировать статистику обращений к серверу. Третий метод может неправильно выбрать кодировку клиента, считая, что все клиенты, работающие под MS Windows, используют кодировку CP-1251.

Учитывая все перечисленные трудности, связанные с русскими кодировками, авторами был предложен новый метод организации доступа к информационным ресурсам Internet через перекодирующий proxy-сервер. Разработанный proxy-сервер может работать в двух режимах: как буферный proxy-сервер между клиентом и WWW-сервером, который перекодирует запросы клиента и ответы сервера с кодировки сервера в кодировку клиента и наоборот, выбор кодировки производится либо клиентом, либо автоматически (если клиентская машина работает “правильно”); как обычный proxy-сервер, который автоматически определяет кодировку, в которой работает сервер (если сервер работает “правильно”), либо исходя из анализа содержания страницы распознает кодировку, в которой работает сервер.

В настоящий момент данная разработка установлена на WWW-серверах Сибирского отделения РАН (http://www-sbras.nsc.ru), Института вычислительных технологий СО РАН (http://www.ict.nsc.ru) и Международного центра новых информационных технологий (http://www.sicnit.ru).

5. Принципы работы перекодирующего proxy-сервера

5.1. Возможности протокола HTTP

Работа перекодирующего proxy-сервера основана на дополнительных возможностях протокола HTTP.

Главная задача протокола HTTP — обеспечить связь между клиентом и сервером, позволяя первым делать запросы, а последним отвечать на них, выдавая запрашиваемые документы вместе с дополнительной META-информацией для того, чтобы клиент мог правильно их интерпретировать. Более того, необходимо, чтобы сами запросы также могли содержать META-информацию, помогающую серверу выполнить запрос наиболее эффективно.

В настоящее время стандартом является версия протокола HTTP/1.1 [3]. HTTP использует заголовки типа MIME для посылки META-информации между сервером и клиентом в запросах и ответах на них. С точки зрения используемых кодировок, самой важной строкой в заголовке ответа является строка Content-type [4], которая сообщает клиенту тип данных, посылаемых сервером: текст на языке HTML (text/html), просто текст (text/plain), изображение в формате GIF (image/gif) и т. д. Но во всех случаях пересылки текстовых

[3]Версия описана описана в документе RFC-2068 (Request for Comments), который можно найти на сервере InterNIC или Сети Internet Новосибирского научного центра.

[4]Кстати, это единственный параметр заголовка, который не имеет значения по умолчанию, и его правильное формирование определяется администратором сервера или клиентом при формировании запросов или отправке электронной почты.

данных правильная интерпретация требует дополнительного указания кодировки символов документа. Для этих целей MIME предлагает использовать параметр charset (сокращение от character set), который добавляется к строке Content-type:

```
Content-Type: text/htm; charset=KOI8-R
```

Почему-то, к сожалению, этот параметр практически никогда не используется в современной практике администраторами русскоязычных серверов. В недалеком прошлом в WWW разрешалось использование только одной кодировки ISO-8859-1, благодаря чему она стала кодировкой по умолчанию в случае, когда charset явно не указан. Американских и европейских администраторов WWW-серверов кодировка ISO-8859-1 вполне устраивает, однако для серверов с русскими текстами мы часто наблюдаем ситуацию, когда просмотрщики (особенно те, которые понимают UNICODE) часто "срываются" при переходе со страницы на страницу на кодовую таблицу ISO-8859-1 (Latin1).

Вместе с тем сейчас, независимо от того, что утверждается в RFC, кодировка ISO-8859-1 уже не является кодировкой по умолчанию. На самом деле такой кодировки просто не существует, так как используются самые разные кодировки без какого-либо указания. Остается только надеяться, что в будущих стандартах HTTP эта проблема займет важное место и параметр charset будет обязательным для всех без исключения. Еще одно поле помогает клиенту правильно интерпретировать полученный текст, а серверу определить, какой язык понимает клиент. С помощью параметра Content-Lanquage можно указать, на каком языке написан документ. В качестве значения указывается двухбуквенное сокращение соответствующего языка (см. RFC-1766). Если документ содержит текст на разных языках, то таких значений может быть несколько, например:

```
Content-Lanquage: ru, en
```

В протоколе HTTP/1.1 существует еще одно средство, позволяющее клиенту сообщать серверу информацию о своей кодировке. Поле Accept-Charset в заголовке запроса может быть использовано для указания кодировок, допустимых в ответе сервера. Если это поле не указано, то предполагается что все кодировки допустимы. Если поле Accept-Charset присутствует, но сервер не может послать ответ клиенту в данной кодировке, сервер должен послать сообщение об ошибке.

Например, если клиент умеет обращаться с документами в кодировке ISO-8859-5 и KOI8-R, то следует указать:

```
Accept-Charset: ISO-8859-5, KOI8-R
```

или

```
Accept-Lanquage: ru, en-US; q-0.8, en; q=0.5,
```

что означает: "Я предпочитаю русский, но могу принять американский английский и, наконец, любые виды английского".

К сожалению, русскоязычные клиенты (серверы, кстати, тоже) практически не используют пока эти возможности. Анализ статистики обращений к серверу Сибирского отделения показал, что за месяц из более чем 340 тыс. обращений к русским страницам только один клиент правильно выдал заголовок запроса.

5.2. Специальные возможности языка HTML

В свою очередь, язык HTML позволяет подсказать клиенту используемую в документе кодировку путем добавления специальной команды (тэга — tag) META в заголовок документа (<META HTTP-EQUIV ... >). В стандарте языка HTML-3.2 утверждается, что

команда META HTTP-EQUIV имеет смысл, только когда документ получен с помощью протокола HTTP. HTTP-серверы должны использовать значение данного атрибута для добавления его в заголовок HTTP-ответа (RFC-822). Этот атрибут не может применяться для задания некоторых особенных полей заголовка. Например, для того чтобы указать, что в документе используется кодировка KOI8-R, можно включить в заголовок документа (то есть в области между <HEAD> и </HEAD>) следующую строку:

```
<META HTTP-EQUIV=>>Content-Type>> CONTENT=>>text/html; charset=KOI8-R>> >
```

По идее, эта META-информация должна относиться только к тому моменту, когда документ выдается сервером. Нет требования к клиенту обращать внимание на эти тэги, однако большинство продолжает их обрабатывать со времен протокола HTTP/0.9. Здесь следует отметить, что просмотрщики корректно обрабатывают эти указатели в случае, если они правильно сконфигурированы, но часть информации выдается просмотрщиком так называемым системным шрифтом (это касается окна status, Java-script'ов и выбора из меню). Системный шрифт определяется локализацией языка в операционной системе и не изменяется просмотрщиком.

Кроме команды META, относящейся ко всему документу, некоторые команды языка HTML имеют дополнительный атрибут charset, который устанавливает кодовую таблицу к текстовой информации, содержащейся внутри команды (этот атрибут почти ни один из просмотрщиков не обрабатывает).

Согласно RFC-2070 ("Internationalization of the Hypertext Markup Lanquage"), все просмотрщики должны использовать следующую систему приоритетов для определения кодировки документа:

1. параметр charset, полученный в заголовке HTTP-ответа из источника месторасположения документа) считается наиболее приоритетным;

2. за ним следует содержимое элемента META в заголовке HTML-документа;

3. наименьшим приоритетом обладает параметр charset, стоящий внутри какого-либо тэга HTML-документа.

Из существующих в данный момент просмотрщиков только Lynx правильно выставляет эти приоритеты. Netscape Navigator правильно показывает документ, полученный из источника, но не сохраняет параметр charset в кэше. Тем самым при взятии документа из собственного кэша он ошибочно полагается на значение META-параметра. Microsoft Internet Explorer вообще считает META-поле более предпочтительным, чем значение charset в заголовке HTTP-ответа.

5.3. Режимы работы proxy-сервера

Разработанный программный модуль может использоваться в трех различных режимах.

Во-первых, как обычный HTTP-proxy, т. е. переадресовывать входящие HTTP-запросы к другим серверам и осуществлять перекодировку полученных от них ответов. Кэширование на диск не производится по той причине, что одна и та же страница может быть запрошена в различных кодировках. Модуль можно настроить так, чтобы запрошенные страницы брались через другие proxy-серверы (например, squid), осуществляющие кэширование, и тем самым в некоторых случаях повышать эффективность работы модуля.

Во-вторых, модуль может выполнять роль предварительного WWW-сервера. При этом получаемые HTTP-запросы будут переадресовываться к какому-либо одному реальному WWW-серверу, а его ответы будут переводиться в нужную кодировку. Благодаря этому

можно держать документы на сервере в одной, удобной для администратора сервера кодировке. При этом реальный сервер может работать на любой платформе. Не требуется никаких изменений ни в самом сервере, ни в документах, хранящихся в нем. Перекодировка осуществляется как в прямом (при передаче данных от сервера к клиенту), так и в обратном направлении (от клиента к серверу, например, при обращении к GGI-скриптам или при обработке активных запросов и форм).

В-третьих, программа может работать как перекодировщик для текстовых протоколов, т.е. протоколов, осуществляющих обмен данными только в текстовом виде, например при работе с почтовым протоколом POP3 (Post-Office Protocol) или при работе с поисковыми системами типа Harvest. Эти протоколы используются для считывания клиентом электронной почты с POP3-сервера или для считывания сервером текстовой информации с WWW-сервера при работе систем контекстного поиска. Настроив proxy-сервер на стандартный POP3-порт (или порт, с которым работает поисковая система), можно получить ситуацию, когда сообщения, получаемые с сервера программой клиента, будут иметь нужную кодировку. Таким образом пользователь будет избавлен от необходимости производить перевод получаемых им сообщений в случае, когда его программа не понимает других кодировок.

5.4. Работа HTTP-proxy

Для определения кодировки документа proxy-сервер использует следующий алгоритм.

1) Если в заголовке HTTP-ответа встречается поле типа `Content-type: text/*; charset=XXX,` то ищем слово XXX в списке синонимов названий кодировок, предлагаемых международной организацией стандартизации. Если такой поиск закончился удачно, то за кодировку документа принимаем главное название из списка синонимов. Далее, в зависимости от того, совпадает ли эта кодировка с заказанной в первом аргументе при запуске программы или нет, производим перекодировку документа с заменой значений этого поля либо пропускаем документ без каких-либо изменений.

2) Если информация, указывающая на кодировку документа, отсутствует, пытаемся определить ее по содержимому самого документа. Алгоритм определения кодировки основан на анализе частот появления символов с кодами большими 127. Если процесс закончился удачно, то при необходимости (то есть в ситуации, когда обнаруженная кодировка не совпадает с запрошенной) перекодируем документ и соответствующим образом устанавливаем поле charset в заголовке HTTP-ответа.

3) Если кодировку определить не удалось, то выдаем документ в том виде, в каком он существует, при этом ничего не меняя.

Таким образом, установив значение “charset” в заголовке HTTP-ответа, мы устанавливаем самый приоритетный параметр и можем быть уверены, что все браузеры, правильно (то есть в соответствии с существующими стандартами и RFC) работающие с перечисленными ранее способами указания кодировки документа, корректно отобразят полученный документ пользователю.

5.5. Работа WWW-сервера

При использовании в этом режиме proxy-сервер может выступать в роли обыкновенного HTTP-сервера, но при этом осуществляя перекодировку документов для клиента. Отличие от режима HTTP-proxy заключается в способе обработки полученных запросов.

Proxy-сервер получает HTTP-запрос и перенаправляет его какому-либо одному указанному в конфигурационном файле реальному HTTP-серверу. Получив ответ, proxy-сервер переводит содержимое в нужную кодировку, определяя исходную кодировку указанным выше способом, и выдает ответ клиенту с указанием соответствующего значения в поле charset. Очевидно, что все обращения к реальному HTTP-серверу будут приходить с адреса, на котором установлен proxy-сервер, то есть в журнале обращений к HTTP-серверу не будет видно реальных адресов клиентов, которые посылали запросы к тем или иным ресурсам. Чтобы обойти эту неприятность, proxy-сервер сам ведет журнал обращений в формате, совместимом с теми, которые используют большинство HTTP-серверов (NSCA, Apache, CERN, Microsoft IIS).

Для удобства пользователей выбирается несколько портов, на которых размещаются вызовы proxy-сервера с различными запрашиваемыми кодировками, например 8000 — KOI8-R, 8001 — CP1251, 8002 — ISO-8859-5 и так далее. На стандартном для протокола HTTP порту (80) размещен реальный HTTP-сервер. При попадании на заглавную страницу пользователю предлагается перейти на один из портов по выбору, если его не устраивает принятая на сервере по умолчанию кодировка (или кодировка, которую попытались определить автоматически). Пользователь переключается на выбранный порт и в дальнейшем уже общается с сервером непосредственно через предварительный proxy-сервер.

При такой организации поддержки различных кодировок отсутствует проблема, связанная с кэшированием содержимого сервера другими proxy-серверами.

Опытная эксплуатация proxy-сервера показала, что в настоящий момент это единственное правильное решение, связанное с организацией WWW-сервера, которое корректно решает проблему множественности русских кодировок.

Поступила в редакцию 24 апреля 1997 г.