Практическая работа № 2
ПЕРВИЧНЫЕ КОДЫ ОБМЕНА ИНФОРМАЦИЕЙ И МЕТОДЫ СЖАТИЯ ДАННЫХ

1. Цель работы:

Углубление фундаментальных знаний в области оптимального кодирования данных в информационных системах, исследование способов построения таблиц кодирования первичных кодов и простейших методов сжатия символьных последовательностей, приобретение практических навыков исследования процессов кодирования информационных сообщений.

2. Теоретический блок:

При начальном кодировании сообщений в процессе подготовки данных на различных носителях ввода, вывода и обработки данных используются так называемые **первичные коды**, которые являются безизбыточными. В настоящее время применяется несколько видов первичных кодов.

Одним из основных первичных кодов является 7-элементный код для обработки информации КОИ-7, созданный на основе международного телеграфного кода МТК-5. Кодовая таблица представляет собой матрицу из 8 столбцов и 16 строк и содержит 128 кодовых комбинаций (КК). Столбцы и строки нумеруются от 0 до 7 и от 0 до 15 соответственно. Комбинация битов кода обозначается b7b6b5b4b3b2b1, где b1 – младший бит КК. Каждая комбинация битов КОИ-7 имеет однозначное соответствие с позицией кодовой таблицы. Позиции определяются в форме дробного числа x/y, где x – номер столбца, y – номер строки.

Кодовая таблица разделена на области, которые предназначены для набора управляющих и графических символов в следующем виде:

1) столбцы 0 и 1 для представления 32 управляющих символов.
2) позиция 2/0 для символа ПРОБЕЛ, который может интерпретироваться как управляющий и как графический символ.
3) позиция 7/15 (последняя) для представления символа ЗАБОЙ.
4) столбцы с 2 по 7 – для представления набора 94 графических символов.

Основная (базисная) таблица кода приведена в таблице 2.1. С целью обеспечения совместимости национальных и проблемно-ориентированных версий введены кодовые позиции, которым не приписываются конкретные графические символы. Для обнаружения ошибок в кодовой комбинации к каждой КК добавляется восьмой контрольный разряд, значение которого равно сумме по модулю 2 всех 7-ми первых битов (проверка на четность).

Международным стандартом для использования в вычислительной технике и информационных системах является таблица ***ASCII*** (*American Standard Code for Information Interchange*), построенная на основе международного телеграфного кода МТК-5. Таблица кодов ASCII делится на две части. Международным стандартом является лишь первая половина таблицы, т.е. символы с номерами от **0** (00000000), до **127** (01111111). В таблице кодировки буквы (прописные и строчные) располагаются в алфавитном порядке, а цифры упорядочены по возрастанию значений. Такое соблюдение лексикографического порядка в расположении символов называется принципом последовательного кодирования алфавита.

Для кодирования символов национальных алфавитов используются расширенные кодировки ASCII**.** В настоящее время существуют пять различных кодировок кириллицы: КОИ8-Р (наличие символов псевдографики), Windows (CP1251-Code Page1251, вместо псевдографики дополнительные кириллические символы украинского, белорусского и болгарского языков), MS-DOS (CP866, наличие символов псевдографики, IBM-PC),

Macintosh и ISO (ISO 8859-5). Из-за этого часто возникают проблемы с переносом русского текста с одного компьютера на другой, из одной программной системы в другую. Для разрешения проблемы совместимости кодировок был разработан новый международный стандарт ***Unicode***.

Юникод — стандарт кодирования символов, включающий в себя знаки почти всех письменных языков мира, в том числе древних и экзотических. В настоящее время стандарт является доминирующим в информационных системах.

Стандарт состоит из двух основных частей: универсального набора символов (*Universal character set,* ***UCS***) и форматы кодировок (*Unicode transformation format,* ***UTF***). Универсальный набор символов ***UCS*** (алфавит кода) перечисляет допустимые по стандарту Юникод символы и присваивает каждому символу код в виде положительного целого числа, записываемого обычно в шестнадцатеричной форме с префиксом U+, например, U+040F. Форматы кодировок определяет способы преобразования кодов символов для передачи их в потоке или в файле.

Коды в стандарте Юникод разделены на несколько областей. Область с кодами от U+0000 до U+007F содержит символы набора ASCII, и коды этих символов совпадают с их кодами в ASCII. Далее расположены области символов других систем письменности, знаки пунктуации и технические символы. Часть кодов зарезервирована для использования в будущем. Юникод имеет несколько форм представления символов в компьютере: ***UTF-8, UTF-16*** *(UTF-16BE, UTF-16LE) и* ***UTF-32*** *(UTF-32BE, UTF-32LE).* (*UTF − Unicode transformation format*).

Первой версией, созданной разработчиками консорциума Юникод, была система кодирования **UTF 32**. Цифра в названии кодировки означает количество бит, которое используется для кодирования одного символа (знака). Основным недостатком ее является то, что из-за кодирования символов четырьмя байтами объем хранимой (передаваемой) информации повышался в 4 раза.

В результате развития Юникода появилась **UTF-16**, которая получилась настолько удачной, что была принята по умолчанию как базовое пространство для всех символов, которые применяются в информационных системах. В этой системе для кодирования одного знака используется два байта.

Но даже эта удачная версия кодировки Юникода не смогла удовлетворить тех, кто писал, например, программы только на английском языке, ибо у них, после перехода от расширенной версии ASCII к UTF-16, объем документов увеличивался в два раза (один байт на один символ в ASCII и два байта на тот же самый символ в ЮТФ-16).

Поэтому консорциумом Unicode была предложена **кодировка переменной длины,** получившая название **UTF-8**. Несмотря на восьмерку в названии, она действительно имеет переменную длину, т.е. каждый символ текста может быть закодирован в последовательность длиной от одного до шести байтов.

На практике же в UTF-8 используется только диапазон от одного до четырех байтов, потому что в алфавитах всего мира, включая самые экзотические, суммарное количество символов не превышает 2^{32}.

В кодировке UTF-8 одним байтом кодируются латинские буквы, цифры и специальные символы. Русские буквы (кириллица) представляются 16-битными (двухбайтными) кодами:

110XXXXX 10XXXXXX,

где **110** − признак двухбайтного кода UTF-8; **10** − признак продолжения многобайтного кода; X − двоичные разряды для размещения кода символа в соответствии с таблицей *UNICODE*. Другие языки кодируются 3-мя или 4-мя байтами. Так, например, грузинские символы кодируются тремя байтами.

Т.о. консорциум Юникод после создания UTF-16 и 8 решил основную проблему, в результате чего в инфокоммуникационных системах **в шрифтах существует единое кодовое пространство**.

Коды, использующие лишь определенную часть всех возможных комбинаций, называются избыточными. Оставшаяся часть комбинаций применяется для обнаружения или исправления ошибок. В этих кодах часть разрядов k используется для кодирования информационной части сообщения, а другая часть – для коррекции ошибок. В теории информации под избыточностью кода *R* понимают отношение числа проверочных разрядов *r* к общей длине кодовой комбинации *n*:

$$R = r / n = (n-k) / n. \quad (2.1)$$

Кодирование информации осуществляют для устранения избыточности — *эффективное кодирование* или для введения дополнительной избыточности — *помехоустойчивое кодирование*, либо с целью защиты (закрытия информации для несанкционированных пользователей) — *шифрование*. Процедуру эффективного кодирования называют также сжатием данных.

Для оценки эффективности процедуры сжатия сообщений используется несколько показателей степени компрессии данных. При оценке эффективности сжатия текстовых сообщений наиболее широко используется коэффициент сжатия $K_{СЖ}$, который характеризует объем сообщения $V_{СЖ}$ (в битах или байтах) на выходе компрессора после сжатия по отношению к исходному объему $V_И$

$$Kсж = Vсж / Vи . \quad (2.2)$$

Часто коэффициент сжатия выражают в процентах. Для этого значение, вычисленное по (2.2), умножается на 100. Очевидно, что при отсутствии сжатия *К*сж = 100 % и уменьшается с повышением эффективности процедуры компрессии.

Оценка уменьшения объема изображений в процессе их сжатия в основном производится с помощью коэффициента компрессии *К*С, который определяется по формуле

$$Кс = V_И / Vсж. \quad (2.3)$$

Он показывает, во сколько раз уменьшился объем исходного сообщения после компрессии. Не трудно заметить, что *Ксж* и *Кс* являются обратными величинами, т. е. *Ксж = 1 / Кс*.

Некоторые авторы для определения степени сжатия используют коэффициент сжатия данных $K_{СД}$, определяемый соотношением

$$K_{СД} = (1 - Vсж / V_И)100\ \%, \quad (2.4)$$

который характеризует объем данных, исключенных из сообщения в процессе его сжатия. При отсутствии эффекта сжатия $K_{СД}$ = 0 %, а в случае максимального сжатия коэффициент $K_{СД}$ приближается к 100%.

В текстовых файлах часто встречаются относительно длинные последовательности одинаковых символов. В первую очередь это символы пробела, дефиса и некоторых специальных знаков. Избыточность таких сообщений может быть существенно сокращена за счет замены группы одинаковых символов последовательностью, состоящей из трёх байтов. Первый является специальным признаком компрессии Sк, индицирующим начало сжатой строки; второй *Ch* -собственно повторяющийся символ и третий *Cn* - счетчик количества одинаковых символов в сжимаемой последовательности. Этот метод получил название "*Run-Length encoding*" *RLE* - кодирование. Например, при поступлении от источника последовательности символов ABCCCCCDEAAAA7C строка сжатых данных приобретает вид ABSкC5DESкA47C, т. е. вместо 15 она занимает объем 12 байтов. Очевидно, что сжимать исходную последовательность целесообразно при длине одинаковых символов в строке не менее 4-х. Максимальное число одинаковых символов ограничивается разрядностью счетчика *Cn* и обычно не превышает 255.

В качестве признака компрессии можно выбрать любую неиспользуемую комбинацию кода КОИ-7 или КОИ-8. Для случаев, когда используются все наборы заданного кода обработки информации, можно применять двойные символы, которые не могут встречаться в тексте (например ЬЬ). В этом случае минимальное количество повторяющихся символов в блоке, которые целесообразно сжимать, равно пяти.

Способ *RLE* достаточно широко использовался в системах архивации до середины 80-х годов, в частности при сжатии псевдографических изображений. В настоящее время он также находит применение при сжатии неподвижных изображений (факсимильные сообщения, файлы *PCX*-формата), а также является составной частью ряда комбинированных способов сжатия.

Если блоки данных преимущественно содержат числовую информацию, то сжатие сообщений может быть достигнуто путем уменьшения числа битов на знак с 7 до 4-х, то есть заменой (*упаковкой*) комбинаций кода КОИ-7 *(ASCII)* на четырёхразрядные. Числа в коде КОИ-7 всегда имеют в трёх старших разрядах комбинацию 011 и поэтому нет необходимости передавать эти биты. Кроме цифр в упакованной форме могут быть переданы знаки (+) и (-), а также десятичная точка (.). Это связано с тем, что младшие тетрады этих символов отличаются от младших тетрад десятичных чисел и при их распаковке могут быть легко преобразованы в соответствующий им эквивалент в *ASCII*-коде. Знак пробела, который часто встречается в цифровых последовательностях, целесообразно кодировать четырёхразрядной комбинацией, состоящей из четырёх единиц 1111, что соответствует младшей тетраде символа (/).

Для того чтобы при распаковке (на приёмной стороне) можно было определить начало упакованной последовательности, используются символ признака сжатия данных *Sк* и счетчик количества упакованных чисел *Cn*, которые размещаются непосредственно перед сжатой последовательностью. Пример фрагмента кадра с упакованной последовательностью десятичных чисел со знаком и десятичной точкой показан на рисунке 2.1. Значок апострофа показывает, что число представлено в упакованном виде.

Sk	*Cn*	2 ’	6 ’	. ‘	3 ‘	/ ‘	. .	/ ‘	5 ‘	7 ‘	4 ‘

Рисунок 2.1- Фрагмент кадра с упакованной последовательностью десятичных чисел.

Коэффициент сжатия упакованной десятичной последовательности зависит от её длины:

$$K_{сж} = V_{УП} / V_И = (n_{Sk} + n_{ch} + 4\,N_Б)\,/\,8\,N_Б\,, \quad (2.5)$$

где $V_{УП}$ и $V_И$ – объем соответственно упакованной и исходной последовательности в битах; $N_В$ – количество байтов исходной числовой последовательности; n_{Sk}, n_{cn} – количество битов для кодирования символа признака сжатия данных и счетчика упакованных чисел соответственно.

Если для представления n_{Sk} и n_{cn} выбирается по одному байту, то коэффициент сжатия рассчитывается по формуле:

$$Kсж = 2 / N_В + 0{,}5\,. \quad (2.6)$$

Отсюда видно, что для того, чтобы Ксж был меньше 1, сжимать следует цифровые последовательности данных не менее пяти байтов. Минимально достижимый коэффициент сжатия равен

$$K_{\text{сж min}} = \lim_{N_B \to \infty} \left(\frac{n_{Sk} + n_{cn}}{8N_B} + 0{,}5\right) = 0{,}5 \quad (2.7)$$

Таким образом, исходная цифровая последовательность может быть сжата до 50% от своего первоначального вида, что эквивалентно повышению эффективной скорости передачи в 2 раза (или соответственно двукратному снижению объема занимаемой памяти).

3. Практический блок:

В качестве установки используется компьютер с установленным файловым менеджером, позволяющий просматривать сообщения, выполненные в различных

кодировках. Вид окна с анализируемым текстом и типами кодировок показан на рисунке 3.1.

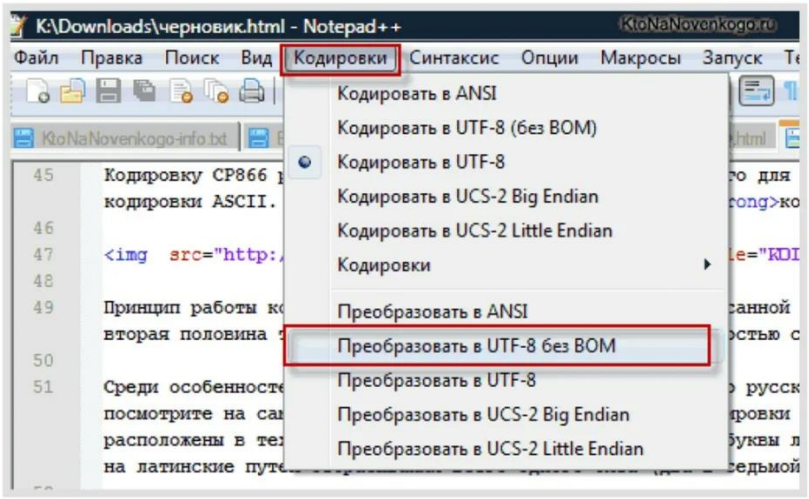

Рисунок 3.1 – Вид окна файлового менеджера с различными кодировками

4. Задание для отчета по лабораторной работе

4.1 Изучить разделы рекомендуемой литературы и конспекта, касающиеся основных понятий теории кодирования, первичных кодов и простейших методов сжатия.
4.2 Набрать в текстовом редакторе (Блокноте) строку произвольного сообщения размером 10-15 символов и сохранить ее в файле.
4.3 Открыть сохраненный файл в режиме просмотра и найти кодировки, в которых происходит правильное отображение текста.
4.4 Записать закодированную строку в 16-ричном коде.
4.5 Найти символы кодируемой строки в таблице CP-1251, выписать их десятичные коды и представить их в двоичном виде. Сравнить эти коды с представлением символов в 16-ричном коде.
4.6 Посмотреть кодируемую строку при кодировке ASCII/DOS, выписать 16-коды символов и сравнить их с кодами соответствующих символов при использовании кодовой страницы CP-1251.
4.7 Выполнить пункт 4.6 при кодовой странице KOI8-R и пояснить причину неверного отображения закодированной строки.
4.8 Вычислить объем изображения, содержащего данные для отображения на экране дисплея с разрешающей способностью 800×600 изображения, в котором на синем фоне в центре экрана располагается красный прямоугольник размером 20×20 пикселей.
4.9 Закодировать содержимое изображения методом RLE и определить объем сжатого файла и рассчитать коэффициент компрессии.
4.10 Составить отчет по выполненной работе.

5. Вопросы для самостоятельного контроля

- Какие существуют виды кодирования и с какой целью они используются?
- С какой целью в таблицы кодирования ASCII и КОИ-7 введены управляющие символы и в каких случаях они используются?
- По какой причине была выполнена разработка стандарта кодирования «Юникод», какие существуют форматы этого кода и в чем их различие?
- Каким образом символы русского алфавита отображаются в кодировке UTF-8?
- Что такое избыточность кода и как она определяется количественно?
- Какие существуют виды кодирования и в каких случаях используется тот или иной вид кодирования?

- Какие показатели используются для оценки сжатия сообщений?
- В каких случаях можно осуществлять сжатие сообщений с частичной потерей информации?
- За счет чего осуществляется сжатие при использовании метода *RLE*?
- Каким образом уменьшают объем сообщений, состоящих из последовательности десятичных цифр?