Московский Авиационный Институт

(Национальный Исследовательский Университет)

Кафедра вычислительной математики и программирования

Реферат на тему «Кодирование информации»

По курсу «Фундаментальная информатика»

I семестр

Выполнил студент
1-го курса, 105-ой группы
Махмудов О. С.

_____

(подпись)

Научный руководитель
Доцент кафедры 806
Сластушенский Ю. В.

_____

(подпись)

Работа защищена
«__»_____ 2019
Оценка_____

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ

Код — это набор условных обозначений (или сигналов) для записи (или передачи) некоторых заранее определенных понятий.

Кодирование информации – это процесс формирования определенного представления информации. В более узком смысле под термином «кодирование» часто понимают переход от одной формы представления информации к другой, более удобной для хранения, передачи или обработки.

Компьютер может обрабатывать только информацию, представленную в числовой форме. Вся другая информация (например, звуки, изображения, показания приборов и т. д.) для обработки на компьютере должна быть преобразована в числовую форму. Например, чтобы перевести в числовую форму музыкальный звук, можно через небольшие промежутки времени измерять интенсивность звука на определенных частотах, представляя результаты каждого измерения в числовой форме. С помощью программ для компьютера можно выполнить преобразования полученной информации, например "наложить" друг на друга звуки от разных источников.

Аналогичным образом на компьютере можно обрабатывать текстовую информацию. При вводе в компьютер каждая буква кодируется определенным числом, а при выводе на внешние устройства (экран или печать) для восприятия человеком по этим числам строятся изображения букв. Соответствие между набором букв и числами называется кодировкой символов.

Как правило, все числа в компьютере представляются с помощью нулей и единиц (а не десяти цифр, как это привычно для людей). Иными словами, компьютеры обычно работают в двоичной системе счисления, поскольку при этом устройства для их обработки получаются значительно более простыми. Ввод чисел в компьютер и вывод их для чтения человеком может осуществляться в привычной десятичной форме, а все необходимые преобразования выполняют программы, работающие на компьютере.

Двоичное кодирование – один из распространенных способов представления информации. В вычислительных машинах, в роботах и станках с числовым программным управлением, как правило, вся информация, с которой имеет дело устройство, кодируется в виде слов двоичного алфавита.

Начиная с конца 60-х годов, компьютеры все больше стали использоваться для обработки текстовой информации, и в настоящее время основная доля

персональных компьютеров в мире (и большая часть времени) занята обработкой именно текстовой информации. Все эти виды информации в компьютере представлены в двоичном коде, т. е. используется алфавит мощностью два (всего два символа 0 и 1). Связано это с тем, что удобно представлять информацию в виде последовательности электрических импульсов: импульс отсутствует (0), импульс есть (1).

Такое кодирование принято называть двоичным, а сами логические последовательности нулей и единиц - машинным языком.

1. ВИДЫ КОДИРОВАНИЯ ИНФОРМАЦИИ

1.1 КОДИРОВАНИЕ ЧИСЕЛ

Система счисления – это способ записи чисел с помощью заданного набора специальных знаков (цифр).

Существуют позиционные и непозиционные системы счисления.

В непозиционных системах для записи числа используется бесконечное множество символов. Примером непозиционной системы счисления может служить римская. Например, для записи числа один используется буква I, два и три выглядят как совокупности символов II, III, но для записи числа пять выбирается новый символ V, шесть - VI, десять - вводится символ X, сто - C, тысяча - M и т.д. Бесконечный ряд чисел потребует бесконечного числа символов для записи чисел. Кроме того, такой способ записи чисел приводит к очень сложным правилам арифметики.

В позиционных системах счисления вес каждой цифры изменяется в зависимости от ее положения (позиции) в последовательности цифр, изображающих число.

Любая позиционная система счисления характеризуется своим основанием. Основание позиционной системы счисления – это количество различных знаков или символов, используемых для изображения цифр в данной системе. За основание системы можно принять любое натуральное число – два, три, четыре и т.д. Следовательно, возможно бесчисленное множество позиционных систем: двоичная, троичная, четверичная и т.д.

Кроме десятичной широко используются системы с основанием, являющимся целой степенью числа *2,* а именно:

· двоичная (используются цифры 0, 1);

· восьмеричная (используются цифры 0, 1, ..., 7);

· шестнадцатеричная (для первых целых чисел от нуля до девяти используются цифры 0, 1, ..., 9, а для следующих чисел - от десяти до пятнадцати – в качестве цифр используются символы A, B, C, D, E, F).

Из всех систем счисления особенно проста и поэтому интересна для технической реализации в компьютерах двоичная система счисления. Компьютеры используют двоичную систему потому, что она имеет ряд преимуществ перед другими системами:

· для ее реализации нужны технические устройства с двумя устойчивыми состояниями (есть ток – нет тока, намагничен – не намагничен и т.п.)

· представление информации посредством только двух состояний надежно и помехоустойчиво

· возможно применение аппарата булевой алгебры для выполнения логических преобразований информации

· двоичная арифметика намного проще десятичной

Для того, чтобы ЭВМ могли анализировать вводимые нами числа, она должна перевести эти числа в двоичную СС и только после этого выполнять операции. При переводе целого десятичного числа в систему с основанием q его необходимо последовательно делить на q до тех пор, пока не останется остаток, меньший или равный q–1. Число в системе с основанием q записывается как последовательность остатков от деления, записанных в обратном порядке, начиная с последнего. Перевод восьмеричных и шестнадцатеричных чисел в двоичную систему очень прост: достаточно каждую цифру заменить эквивалентной ей двоичной триадой (тройкой цифр) или тетрадой (четверкой цифр).

Для представления знаковых целых чисел в ЭВМ используются три способа:

1)*прямой* код;

2)*обратный* код;

3)*дополнительный* код.

Все три способа используют самый левый (старший) разряд битового набора длины *k* для кодирования знака числа: знак «плюс» кодируется нулем, а «минус» — единицей. Остальные $k - 1$ разрядов (называемые *мантиссой* или цифровой частью) используются для представления абсолютной величины числа.

Положительные целые числа (и число 0)

Положительные числа в прямом, обратном и дополнительном кодах изображаются одинаково — цифровая часть содержит двоичную запись числа, в знаковом разряде содержится 0. (Приложение 1)

Отрицательные целые числа

- Прямой код отрицательных чисел

В знаковый разряд помещается цифра 1, а в разряды цифровой части числа — двоичный код его абсолютной величины. (Приложение 2)

- Обратный код отрицательных чисел

Получается инвертированием всех цифр двоичного кода абсолютной величины числа, включая разряд знака: нули заменяются единицами, а единицы — нулями. (Приложение 3)

- Дополнительный код отрицательных чисел

Получается образованием обратного кода с последующим прибавлением единицы к его младшему разряду. В дополнительном коде представление нуля единственно, в отличии от прямого и обратного кода.(Приложение 4)

1.2 КОДИРОВАНИЕ ГРАФИЧЕСКОЙ И ЗВУКОВОЙ ИНФОРМАЦИИ

Важным этапом кодирования графического изображения является разбиение его на дискретные элементы (дискретизация). Основными способами представления графики для ее хранения и обработки с помощью компьютера являются растровые и векторные изображения.

Векторное изображение представляет собой графический объект, состоящий из элементарных геометрических фигур (чаще всего отрезков и дуг). Положение этих элементарных отрезков определяется координатами точек и величиной радиуса. Для каждой линии указывается двоичные коды типа линии (сплошная, пунктирная, штрихпунктирная), толщины и цвета.

Растровое изображение представляет собой совокупность точек (пикселей), полученных в результате дискретизации изображения в соответствии с матричным принципом.

Матричный принцип кодирования графических изображений заключается в том, что изображение разбивается на заданное количество строк и столбцов. Затем каждый элемент полученной сетки кодируется по выбранному правилу.

Pixel (picture element - элемент рисунка) - минимальная единица изображения, цвет и яркость которой можно задать независимо от остального изображения. В соответствии с матричным принципом строятся изображения, выводимые на принтер, отображаемые на экране дисплея, получаемые с помощью сканера.

Качество изображения будет тем выше, чем "плотнее" расположены пиксели, то есть чем больше разрешающая способность устройства, и чем точнее закодирован цвет каждого из них.

Для черно-белого изображения код цвета каждого пикселя задается одним битом. Если рисунок цветной, то для каждой точки задается двоичный код ее цвета.

Поскольку и цвета кодируются в двоичном коде, то если, например, вы хотите использовать 16-цветный рисунок, то для кодирования каждого пикселя вам потребуется 4 бита (16=24), а если есть возможность использовать 16 бит (2 байта) для кодирования цвета одного пикселя, то вы можете передать тогда 216 = 65536 различных цветов. Использование трех байтов (24 битов) для кодирования цвета одной точки позволяет отразить 16777216 (или около 17 миллионов) различных оттенков цвета - так называемый режим “истинного цвета” (True Color).

Различные звуковые карты могут обеспечить как 8-, так и 16-битные выборки. При замене непрерывного звукового сигнала его дискретным представлением в виде ступенек 8-битные карты позволяют закодировать 256 различных уровней дискретизации звукового сигнала, соответственно 16-битные - 65 536 уровней.

1.3 КОДИРОВАНИЕ ТЕКСТОВОЙ ИНФОРМАЦИИ

Для представления символов в числовой форме был предложен метод кодирования, получивший в дальнейшем широкое распространение и для других видов представления нечисловых данных (звуков, изображений и др.). Под алфавитом компьютерной системы понимают совокупность вводимых и отображаемых символов. Алфавит компьютерной системы включает в себя арабские цифры, буквы латинского алфавита, знаки препинания, специальные символы и знаки, буквы национального алфавита, символы псевдографики - растры, прямоугольники, одинарные и двойные рамки, стрелки.

Нажатие алфавитно-цифровой клавиши на клавиатуре приводит к тому, что в компьютер посылается сигнал в виде двоичного числа, представляющего собой одно из значений кодовой таблицы. Кодовая таблица - это внутреннее представление символов в компьютере. Во всем мире в качестве стандарта принята таблица ASCII, о ней поговорим чуть позже.

Для хранения двоичного кода одного символа выделен 1 байт = 8 бит. Учитывая, что каждый бит принимает значение 1 или 0, количество возможных сочетаний единиц и нулей равно $2^8 = 256$. Следовательно, с помощью 1 байта можно получить 256 разных двоичных кодовых комбинаций и отобразить с их помощью 256 различных символов. Эти коды и составляют таблицу ASCII.

Например, при нажатии клавиши с буквой «Н» в память компьютера записывается код 01001000. При выводе буквы «Н» на экран компьютер выполняет декодирование - на основании этого двоичного кода строится изображение символа.

Примечание. Цифры кодируются по стандарту ASCII в двух случаях - при вводе-выводе и когда они встречаются в тексте. Если цифры участвуют в вычислениях, то осуществляется их преобразование в двоичный код по правилам перевода чисел из одной системы счисления в другую.

Для сравнения рассмотрим представление числа 27_{10} для двух вариантов кодирования.

При использовании в тексте это число потребует для своего представления 2 байта, поскольку каждая цифра будет представлена своим кодом в соответствии с таблицей ASCII . В двоичной системе – 00110010 00110111.

При использовании в вычислениях код этого числа будет получен по специальным правилам перевода и представлен в виде 8-разрядного двоичного числа 00011011, на что потребуется 1 байт.

2. ТАБЛИЦЫ ДЛЯ КОДИРОВАНИЯ ТЕКСТОВОЙ ИНФОРМАЦИИ

В вычислительных машинах символы не могут храниться иначе, как в виде последовательностей бит (как и числа). Для передачи символа и его корректного отображения ему должна соответствовать уникальная последовательность нулей и единиц. Для этого были разработаны

таблицы кодировок.

Количество символов, которые можно задать последовательностью бит длины n, задается простой формулой $C(n)=2n$C(n)=2n. Таким образом, от нужного количества символов напрямую зависит количество используемой памяти.

2.1 КОДИРОВКА ASCII и КОИ8

ASCII *(**A**merican **s**tandard **c**ode for **i**nformation **i**nterchange)* — первая кодировка, пригодная для работы с текстом. Помимо маленьких букв английского алфавита и служебных символов, содержит большие буквы английского языка, цифры, знаки препинания и другие символы. Таблица была разработана и стандартизована в США, в 1963 году. (Приложение 5)

Изначально (1963 год) ASCII была разработана для кодирования символов, коды которых помещались в 7 бит (128 символов); при этом старший 7-й бит (нумерация с нуля) использовался для контроля ошибок, возникших при передаче данных. Со временем — кодировка была расширена до 256 символов, коды первых 128 символов не изменились. ASCII стала восприниматься как половина 8-битной кодировки, а «расширенной ASCII» называли ASCII с задействованным 8-м битом. Это первая кодировка, в которой можно было использовать символы национальных алфавитов.

КОИ-8 (код обмена информацией, 8 бит) — восьмибитовая кодовая страница, совместимая с ASCII. Разработана для кодирования букв кириллических алфавитов. Была широко распространена как основная русская кодировка в Unix-совместимых ОС и в электронной почте.

Разработчики КОИ-8 поместили символы русского алфавита в верхней части кодовой таблицы таким образом, что позиции символов кириллицы соответствуют их фонетическим аналогам в английском алфавите из нижней части таблицы. Это означает, что если в тексте, написанном в КОИ-8, убрать восьмой бит каждого символа, то получится «читаемый» текст, подобный транслиту. Например, слова «Русский Текст» превратятся в «rUSSKIJ tEKST». Из-за этого символы кириллицы расположены не в алфавитном порядке.

Есть несколько вариантов кодировки КОИ-8 для различных кириллических алфавитов, расширяющие определённые коды (общий диапазон 192—255 с 32 русскими буквами в двух регистрах остаётся неизменным во всех вариантах). Русский алфавит описывается в кодировке KOI8-R, украинский — в KOI8-U, таджикский — в KOI8-T.

2.2 ОСНОВНАЯ И АЛЬТЕРНАТИВНАЯ КОДИРОВКИ

Основная кодировка была принята в 1987 г. взамен КОИ-8, однако использовалась мало. Основную кодировку поддерживало только оборудование и программное обеспечение, производившееся в СССР (ЕС ПЭВМ, Лексикон), а также некоторые принтеры Epson. На базе основной кодировки была создана ISO 8859-5, но и она не нашла широкого применения. Гораздо более популярной оказалась альтернативная кодировка (с тем же набором символов, но в другом порядке).

Альтернативная кодировка — основанная на CP437 кодовая страница, где все специфические европейские символы во второй половине заменены на кириллицу, оставляя псевдографические символы нетронутыми. Окончательным стандартом стала кодировка IBM CP866. В этой кодировке записываются имена файлов в системе FAT. Поныне является популярной стандартной кодировкой Microsoft в среде DOS и OS/2, используется в консоли русифицированных систем семейства Windows NT. Вне среды MS-DOS в Microsoft Windows заменена стандартной кодировкой CP1251, а в операционных системах Windows NT и следующих за ней - кодировкой Юникод.

Основная и альтернативная кодировки отличаются от ASCII способами добавления русских букв в расширенную часть таблицы (8-й бит равен 1) и размещением знаков псевдографики. В основной кодировке русские буквы размещены подряд и без разрывов. В альтернативной кодировке заглавные буквы размещены подряд, а малые – разбиты на два поддиапазона, между которыми для совместимости с кодировкой IBM размещена псевдографика.

2.3 КОДИРОВКА ISO 8859 и ISO8859-5

ISO 8859 - семейство ASCII-совместимых кодовых страниц, разработанное совместными усилиями ISO и IEC. По состоянию на 2006 год, это семейство состояло из 15 кодовых страниц.

Поскольку кодировки ISO 8859 разрабатывались как средства для обмена информацией, а не как средства обеспечения высококачественной типографики, то в них не включены такие символы, как парные кавычки, тире различной длины, лигатуры и т. П. (хотя там всё же присутствуют такие символы, как неразрывный пробел и символ мягкого переноса). Зато довольно много места (область 0x80—0x9F) зарезервировано под «верхние управляющие символы», предназначенные для управления терминалами.

Поскольку различные страницы ISO 8859 разрабатывались совместно, они обладают некоторой взаимной совместимостью. Например, все семь символов расширенной латиницы, используемые в немецком языке, стоят на одинаковых позициях во всех кодовых страницах, включающих эти символы. Страницы Latin-1—Latin-4 обладают ещё большей степенью совместимости: каждый символ, представленный в любой из этих страниц, стоит в них на одинаковых позициях.

Кодировки серии ISO 8859 применяются главным образом на юниксоподобных системах, а также для кодирования веб-страниц (поскольку большинство веб-серверов использует UNIX).

ISO 8859-5 — 8-битная кодовая страница из семейства кодовых страниц стандарта ISO-8859 для представления кириллицы. ISO 8859-5 была создана на базе «основной кодировки». Имеются буквы многих языков, использующих кириллицу, однако в целом ISO 8859-5 — не очень удобная кодировка, поскольку в ней отсутствуют многие нужные символы, такие как тире (—), кавычки-ёлочки («»), градус (°) и др. Нет также буквы Ґ, используемой в украинской письменности. (Приложение 6)

2.4 КОДИРОВКА CP-1251(Windows-1251)

Windows-1251 — набор символов и кодировка, являющаяся стандартной 8-битной кодировкой для русских версий Microsoft Windows до 10-й версии. В прошлом пользовалась довольно большой популярностью. Была создана на базе кодировок, использовавшихся в ранних «самопальных» русификаторах Windows в 1990—1991 гг. совместно представителями «Параграфа», «Диалога» и российского отделения Microsoft.

Windows-1251 выгодно отличается от других 8-битных кириллических кодировок (таких как CP866, KOI8-R и ISO 8859-5) наличием практически всех символов, использующихся в русской типографике для обычного текста (отсутствует только значок ударения); Она также содержит все символы для других славянских языков: украинского, белорусского, сербского, македонского и болгарского. (Приложение 7)

Windows-1251 имеет два недостатка:

- строчная буква «я» имеет код 0xFF (255 в десятичной системе). Она является «виновницей» ряда неожиданных проблем в программах без поддержки чистого 8-го бита, а также (гораздо более частый случай) использующих этот код как служебный (в CP437 он обозначает

«неразрывный пробел», в Windows-1252 — ÿ, оба варианта практически не используются; число же `-1`, в дополнительном коде длиной 8 бит, представляющееся числом `255`, часто используется в программировании как специальное значение).

- отсутствуют символы псевдографики, имеющиеся в CP866 и KOI8 (хотя для самих Windows, для которых она предназначена, в них не было нужды, это делало несовместимость двух использовавшихся в них кодировок заметнее).

Также как недостаток может рассматриваться отдельное расположение буквы «ё», тогда как остальные символы расположены строго в алфавитном порядке.

2.5 КОДИРОВКА UNICODE

Unicode — стандарт кодирования символов, включающий в себя знаки почти всех письменных языков мира. В настоящее время стандарт является доминирующим в Интернете.

Стандарт предложен в 1991 году некоммерческой организацией «Консорциум Юникода» (англ. Unicode Consortium, Unicode Inc.). Применение этого стандарта позволяет закодировать очень большое число символов из разных систем письменности: в документах, закодированных по стандарту Юникод, могут соседствовать китайские иероглифы, математические символы, буквы греческого алфавита, латиницы и кириллицы, символы музыкальной нотной нотации, при этом становится ненужным переключение кодовых страниц.

Стандарт состоит из двух основных частей: универсального набора символов (англ. Universal character set, UCS) и семейства кодировок (англ. Unicode transformation format, UTF). Универсальный набор символов перечисляет допустимые по стандарту Юникод символы и присваивает каждому символу код в виде неотрицательного целого числа, записываемого обычно в шестнадцатеричной форме с префиксом `U+`, например, `U+040F`. Семейство кодировок определяет способы преобразования кодов символов для передачи 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[7]. Под символы кириллицы выделены области знаков с кодами от U+0400 до U+052F, от U+2DE0 до U+2DFF, от U+A640 до U+A69F.

Юникод имеет несколько форм представления (англ. Unicode transformation format, UTF): UTF-8, UTF-16 (UTF-16BE, UTF-16LE) и UTF-32 (UTF-32BE, UTF-32LE). Была разработана также форма представления UTF-7 для передачи по семибитным каналам, но из-за несовместимости с ASCII она не получила распространения и не включена в стандарт. 1 апреля 2005 года были предложены две шуточные формы представления: UTF-9 и UTF-18 (RFC 4042).

В Microsoft Windows NT и основанных на ней системах Windows 2000 и Windows XP в основном используется форма UTF-16LE. В UNIX-подобных операционных системах GNU/Linux, BSD и Mac OS X принята форма UTF-8 для файлов и UTF-32 или UTF-8 для обработки символов в оперативной памяти.

Punycode — другая форма кодирования последовательностей Unicode-символов в так называемые ACE-последовательности, которые состоят только из алфавитно-цифровых символов, как это разрешено в доменных именах.

UTF-8 — представление Юникода, обеспечивающее наибольшую компактность и обратную совместимость с 7-битной системой ASCII; текст, состоящий только из символов с номерами меньше 128, при записи в UTF-8 превращается в обычный текст ASCII и может быть отображён любой программой, работающей с ASCII; и наоборот, текст, закодированный 7-битной ASCII может быть отображён программой, предназначенной для работы с UTF-8. Остальные символы Юникода изображаются последовательностями длиной от 2 до 4 байт, в которых первый байт всегда имеет маску `11xxxxxx`, а остальные — `10xxxxxx`. В UTF-8 не используются суррогатные пары.

Формат UTF-8 был изобретён 2 сентября 1992 года Кеном Томпсоном и Робом Пайком и реализован в ОС Plan 9[64]. Сейчас стандарт UTF-8 официально закреплён в документах RFC 3629 и ISO/IEC 10646 Annex D.

UTF-16 — кодировка, позволяющая записывать символы Юникода в диапазонах U+0000...U+D7FF и U+E000...U+10FFFF (общим количеством 1 112 064). При этом каждый символ записывается одним или двумя словами (суррогатная пара). Кодировка UTF-16 описана в приложении Q к международному стандарту ISO/IEC 10646, а также ей посвящён документ IETF RFC 2781 под названием «UTF-16, an encoding of ISO 10646».

UTF-32 — способ представления Юникода, при котором каждый символ занимает ровно 4 байта. Главное преимущество UTF-32 перед кодировками переменной длины заключается в том, что символы Юникод в ней непосредственно индексируемы, поэтому найти символ по номеру его позиции в файле можно чрезвычайно быстро, и получение любого символа n-й позиции при этом является операцией, занимающей всегда одинаковое время. Это также делает замену символов в строках UTF-32 очень простой. Напротив, кодировки с переменной длиной требуют последовательного доступа к символу n-й

позиции, что может быть очень затратной по времени операцией. Главный недостаток UTF-32 — это неэффективное использование пространства, так как для хранения любого символа используется четыре байта. Символы, лежащие за пределами нулевой (базовой) плоскости кодового пространства, редко используются в большинстве текстов. Поэтому удвоение, в сравнении с UTF-16, занимаемого строками в UTF-32 пространства, зачастую не оправдано.

Юникод включает практически все современные письменности, в том числе:

- арабскую,
- армянскую,
- бенгальскую,
- бирманскую,
- глаголицу,
- греческую,
- грузинскую,
- деванагари,
- еврейскую,
- кириллицу,
- китайскую (китайские иероглифы активно используются в японском языке, а также изредка в корейском),
- коптскую,
- кхмерскую,
- латинскую,
- тамильскую,
- корейскую (хангыль),
- чероки,
- эфиопскую,
- японскую (которая включает в себя, кроме слоговой азбуки, ещё и китайские иероглифы)

и другие.

Проблемы Unicode:

В Юникоде английское «a» и польское «a» — один и тот же символ. Точно так же одним и тем же символом (но отличающимся от «a» латинского) считаются русское «а» и сербское «а». Такой принцип кодирования не универсален; по-видимому, решения «на все случаи жизни» вообще не может существовать.

- Тексты на китайском, корейском и японском языках имеют традиционное написание сверху вниз, начиная с правого верхнего угла. Переключение горизонтального и вертикального написания для этих языков

не предусмотрено в Юникоде — это должно осуществляться средствами языков разметки или внутренними механизмами текстовых процессоров.

- Наличие или отсутствие в Юникоде разных начертаний одного и того же символа в зависимости от языка. Нужно следить, чтобы текст всегда был правильно помечен как относящийся к тому или другому языку.

  Так, китайские иероглифы могут иметь разные начертания в китайском, японском (кандзи) и корейском (ханча), но при этом в Юникоде обозначаются одним и тем же символом (так называемая CJK-унификация), хотя упрощённые и полные иероглифы всё же имеют разные коды.

  Аналогично, русский и сербский языки используют разное начертание курсивных букв п и т (в сербском они выглядят как и и ш, см. сербский курсив).

- Перевод из строчных букв в заглавные тоже зависит от языка. Например: в турецком существуют буквы İi и Iı — таким образом, турецкие правила изменения регистра конфликтуют с английскими, которые предписывают «i» переводить в «I». Подобные проблемы есть и в других языках — например, в канадском диалекте французского языка регистр переводится немного не так, как во Франции.
- Даже с арабскими цифрами есть определённые типографские тонкости: цифры бывают «прописными» и «строчными», пропорциональными и моноширинными — для Юникода разницы между ними нет. Подобные нюансы остаются за программным обеспечением.

Некоторые недостатки связаны не с самим Юникодом, а с возможностями обработчиков текста.

- Файлы нелатинского текста в Юникоде всегда занимают больше места, так как один символ кодируется не одним байтом, как в различных национальных кодировках, а последовательностью байтов (исключение составляет UTF-8 для языков, алфавит которых укладывается в ASCII, а также наличие в тексте символов двух и более языков, алфавит которых не укладывается в ASCII[67]). Файл шрифта, необходимый для отображения всех символов таблицы Юникод, занимает сравнительно много места в памяти и требует бóльших вычислительных ресурсов, чем шрифт только одного национального языка пользователя. С увеличением мощности компьютерных систем и удешевлением памяти и дискового пространства эта проблема становится всё менее существенной; тем не менее, она остаётся актуальной для портативных устройств, например, для мобильных телефонов.
- Хотя поддержка Юникода реализована в наиболее распространённых операционных системах, до сих пор не всё прикладное

программное обеспечение поддерживает корректную работу с ним. В частности, не всегда обрабатываются метки порядка байтов (BOM) и плохо поддерживаются диакритические символы. Проблема является временной и есть следствие сравнительной новизны стандартов Юникода (в сравнении с однобайтовыми национальными кодировками).

- Производительность всех программ обработки строк (в том числе и сортировок в БД) снижается при использовании Юникода вместо однобайтовых кодировок.

ЗАКЛЮЧЕНИЕ

Таким образом можно сделать вывод, что кодирование информации в компьютере - это очень важный процесс. Поскольку компьютеры должны быть быстрыми и с высокой производительностью, вся информация в них представлена ввиде двоичного кода, поэтому любая информация должна быть перекодирована в двоичный код (машинный код). Цифровая информация кодируется с помощью расчетов в системах счисления, графическая и звуковая, с помощью формул, а с кодированием текстовой информации не все так просто. Текстовая информация кодируется с помощью специальных таблиц кодирования, в которых каждый символ соответствует своему двоичному коду. С развитием ЭВМ и компьютеров в частности нужно было развивать и кодировки, поэтому развитие кодировок началось с середины XX века и продолжается до наших дней, так как появляется больше символов и письменных народных языков.

СПИСОК ЛИТЕРАТУРЫ

http://mirznanii.com/a/112358/kodirovanie-informatsii

https://studopedia.su/6_40451_sistemi-schisleniya-predstavlenie-chisel-v-evm.html

https://studfiles.net/preview/4695340/

https://cyberpedia.su/8xecfc.html

https://ru.wikipedia.org/wiki/ASCII

http://mylektsii.ru/3-63631.html

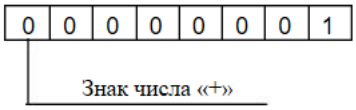
https://ru.wikipedia.org/wiki/ISO_8859

https://ru.wikipedia.org/wiki/Юникод

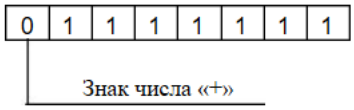
ПРИЛОЖЕНИЯ

1)

Число $1_{10} = 1_2$

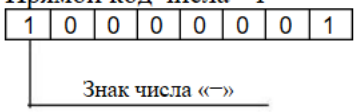


Число $127_{10} = 1111111_2$

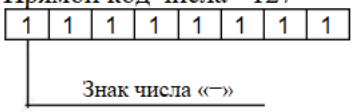


2)

Прямой код числа −1



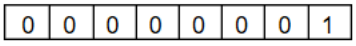
Прямой код числа −127



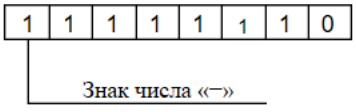
3)

Число −1

Код модуля числа:

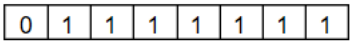


Обратный код числа:



Число −127

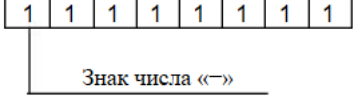
Код модуля числа:



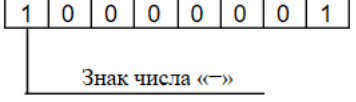
Обратный код числа:

4)

Дополнительный код числа −1



Дополнительный код числа −127



5)

32	пробел	48	0	64	@	80	P	96	`	112	p
33	!	49	1	65	A	81	Q	97	a	113	q
34	"	50	2	66	B	82	R	98	b	114	r
35	#	51	3	67	C	83	S	99	c	115	s
36	$	52	4	68	D	84	T	100	d	116	t
37	%	53	5	69	E	85	U	101	e	117	u
38	&	54	6	70	F	86	V	102	f	118	v
39	'	55	7	71	G	87	W	103	g	119	w
40	(	56	8	72	H	88	X	104	h	120	x
41	)	57	9	73	I	89	Y	105	i	121	y
42	*	58	:	74	J	90	Z	106	j	122	z
43	+	59	;	75	K	91	[	107	k	123	{
44	,	60	<	76	L	92	\	108	l	124	|
45	-	61	=	77	M	93	]	109	m	125	}
46	.	62	>	78	N	94	^	110	n	126	~
47	/	63	?	79	O	95	_	111	o	127	

6)

128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143
144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159
nbsp 160	Ё 161	Ђ 162	Ѓ 163	Є 164	S 165	I 166	Ї 167	J 168	Љ 169	Њ 170	Ћ 171	Ќ 172	shy 173	Ў 174	Џ 175
А 176	Б 177	В 178	Г 179	Д 180	Е 181	Ж 182	З 183	И 184	Й 185	К 186	Л 187	М 188	Н 189	О 190	П 191
Р 192	С 193	Т 194	У 195	Ф 196	Х 197	Ц 198	Ч 199	Ш 200	Щ 201	Ъ 202	Ы 203	Ь 204	Э 205	Ю 206	Я 207
а 208	б 209	в 210	г 211	д 212	е 213	ж 214	з 215	и 216	й 217	к 218	л 219	м 220	н 221	о 222	п 223
р 224	с 225	т 226	у 227	ф 228	х 229	ц 230	ч 231	ш 232	щ 233	ъ 234	ы 235	ь 236	э 237	ю 238	я 239
№ 240	ё 241	ђ 242	ѓ 243	є 244	s 245	і 246	ї 247	j 248	љ 249	њ 250	ћ 251	ќ 252	§ 253	ў 254	џ 255

7)

128 Ђ	144 ђ	160	176 °	192 А	208 Р	224 а	240 р
129 Ѓ	145 ‘	161 Ў	177 ±	193 Б	209 С	225 б	241 с
130 ‚	146 ’	162 ў	178 І	194 В	210 Т	226 в	242 т
131 ѓ	147 “	163 Ј	179 і	195 Г	211 У	227 г	243 у
132 „	148 ”	164 ¤	180 ґ	196 Д	212 Ф	228 д	244 ф
133 …	149 •	165 Ґ	181 µ	197 Е	213 Х	229 е	245 х
134 †	150 –	166 ¦	182 ¶	198 Ж	214 Ц	230 ж	246 ц
135 ‡	151 —	167 §	183 ·	199 З	215 Ч	231 з	247 ч
136	152	168 Ё	184 ё	200 И	216 Ш	232 и	248 ш
137 ‰	153 ™	169 ©	185 №	201 Й	217 Щ	233 й	249 щ
138 Љ	154 љ	170 Є	186 є	202 К	218 Ъ	234 к	250 ъ
139 ‹	155 ›	171 «	187 »	203 Л	219 Ы	235 л	251 ы
140 Њ	156 њ	172 ¬	188 ј	204 М	220 Ь	236 м	252 ь
141 Ќ	157 ќ	173 -	189 S	205 Н	221 Э	237 н	253 э
142 Ћ	158 ћ	174 ®	190 s	206 О	222 Ю	238 о	254 ю
143 Џ	159 џ	175 Ї	191 ї	207 П	223 Я	239 п	255 я