## Учреждение образования

## Белорусский государственный технологический университет

Кафедра полиграфического оборудования и

системы обработки информации

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 3**

по дисциплине «Стандартизация и сертификация

информационных систем и технологий»

Тема

«Применение стандартов кодирования

текстовой информации»

Выполнил студент:

Миневич Кристина

ф-та ИТ 4 к. 4 гр.

Проверил

кандидат технических наук

Сулим П.Е.

Отчет по лабораторной работе

защищен с отметкой баллов

Минск 2024

**Лабораторная работа № 3**

**ПРИМЕНЕНИЕ СТАНДАРТОВ КОДИРОВАНИЯ**

**ТЕКСТОВОЙ ИНФОРМАЦИИ**

***Цель* *работы*:**

Представление текстовой информации – разобраться со стандартами и ответить на вопросы.

**Описание работы:**

**ASCII** — таблицы кодировок, в которых содержатся основные символы (английский алфавит, цифры, знаки препинания, символы национальных алфавитов (свои для каждого региона), служебные символы) и длина кода каждого символа бит.

**Бит:**

* **ASCII7** — 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.

**Кодировки стандарта ASCII (бит):**

* **ASCII** (***ASCII*** — **A**merican **S**tandard **C**ode for **I**nformation **I**nterchange) - первая кодировка, в которой стало возможно использовать символы национальных алфавитов.
* **КОИ8-R** — первая русская кодировка. Символы кириллицы расположены не в алфавитном порядке. Их разместили в верхнюю половину таблицы так, чтобы позиции кириллических символов соответствовали их фонетическим аналогам в английском алфавите. Это значит, что даже при потере старшего бита каждого символа, например, при проходе через устаревший семибитный модем, текст остается "читаемым".
* **CP866** — русская кодировка, использовавшаяся на компьютерах IBM в системе DOS.
* **Windows-1251** —русская кодировка, использовавшаяся в русскоязычных версиях операционной системы Windows в начале 90-х годов. Кириллические символы идут в алфавитном порядке. Содержит все символы, встречающиеся в типографике обычного текста (кроме знака ударения).

Стандарты для следующих *coded character sets*:

* *ASCII* *character set*

ASCII означает «Американский стандартный код для обмена информацией». Он был разработан в начале 60-х годов как стандартный набор символов для компьютеров и электронных устройств. ASCII — это 7-битный набор символов, содержащий 128 символов. Он содержит цифры от 0 до 9, заглавные и строчные английские буквы от A до Z и некоторые специальные символы. Наборы символов, используемые в современных компьютерах, в HTML и в Интернете, основаны на ASCII.

* *ANSI (****A****merican* ***N****ational* ***S****tandard* ***I****nstitute) character set*

Набор из 217 символов ANSI, также известный как Windows-1252, был стандартом для основных шрифтов, поставляемых с американскими версиями Microsoft Windows вплоть до Windows 95 и Windows NT 4 включительно. За время существования этих двух продуктов Microsoft добавила символ валюты евро, доведя количество символов до 218, и представил новый базовый набор общеевропейских шрифтов, содержащий набор символов WGL4 (Windows Glyph List 4) с 652 символами.

* *Universal Coded Character Set*

**Unicode** расшифровывают как **Universal** **Coded** **Character** **Set** (UCS), и у него есть официальное обозначение ISO/IEC 10646. Но обычно все используют название **Unicode**. Этот стандарт помог решить проблемы, возникавшие из-за кодировки и кодовых страниц. Он содержит множество кодовых пунктов (кодовых точек), присвоенных символам из языков и культур со всего мира.

* 1. **Освоить понятие *Unicode* кодеки: *UTF*-8, *UTF*-16, *UTF*-32.**

**UTF-8** является юникод-кодировкой переменной длины, с помощью которой можно представить любой символ юникода.

Первым делом надо сказать, что структурной (атомарной) единицей этой кодировки является **байт**. То, что кодировка переменной длинны, значит, что один символ может быть закодирован разным количеством структурных единиц кодировки, то есть **разным количеством байтов**. Так, например, латиница кодируется одним байтом, а кириллица двумя байтами.

**UTF-16** также является кодировкой переменной длинны. Главное ее отличие от UTF-8 состоит в том, что структурной единицей в ней является не один, а **два байта**. То есть в кодировке UTF-16 любой символ юникода может быть закодирован либо двумя, либо **четырьмя байтами**. Исходя из этого любой символ юникода в кодировке UTF-16 может быть закодирован либо одной кодовой парой, либо двумя.

Начнем с символов, которые кодируются одной кодовой парой. Легко посчитать что таких символов может быть 65 535 (2в16), что полностью совпадает с базовым блоком юникода. Все символы, находящиеся в этом блоке юникода в кодировке UTF-16, будут закодированы одной кодовой парой (двумя байтами), тут все просто.

***символ «o» (латиница) — 00000000 01101111***

***символ «M» (кириллица) — 00000100 00011100***

**UTF-32** в [информатике](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0) ***–*** один из [способов кодирования](https://ru.wikipedia.org/wiki/%D0%9D%D0%B0%D0%B1%D0%BE%D1%80_%D1%81%D0%B8%D0%BC%D0%B2%D0%BE%D0%BB%D0%BE%D0%B2) символов [Юникода](https://ru.wikipedia.org/wiki/%D0%AE%D0%BD%D0%B8%D0%BA%D0%BE%D0%B4), использующий для кодирования любого символа ровно **32 бита** – **4 байта**. Остальные кодировки, UTF-8 и UTF-16, используют для представления символов переменное число байтов. Символ UTF-32 является прямым представлением его кодовой позиции ([Code point](https://en.wikipedia.org/wiki/Code_point) (англ.)[русск.](https://ru.wikipedia.org/w/index.php?title=Code_point&action=edit&redlink=1)).

Главное преимущество UTF-32 перед кодировками переменной длины заключается в том, что **символы Юникод непосредственно индексируемы**. Получение n-ой кодовой позиции является операцией, занимающей одинаковое время. Напротив, коды с переменной длиной требует последовательного доступа к n-ой кодовой позиции. Это делает замену символов в строках UTF-32 простой, для этого используется целое число в качестве индекса, как обычно делается для строк [ASCII](https://ru.wikipedia.org/wiki/ASCII).

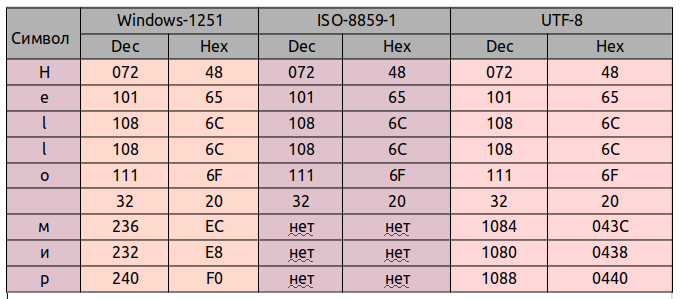
**Главный недостаток UTF-32 — это неэффективное использование пространства, так как для хранения символа используется четыре байта.** Символы, лежащие за пределами нулевой (базовой) плоскости кодового пространства, редко используются в большинстве текстов. Поэтому удвоение, в сравнении с UTF-16, занимаемого строками в UTF-32 пространства, не оправдано.

Хотя использование неменяющегося числа байтов на символ удобно, но не настолько, как кажется. Операция усечения строк реализуется легче в сравнении с UTF-8 и UTF-16. Но это не делает более быстрым нахождение конкретного смещения в строке, так как смещение может вычисляться и для кодировок фиксированного размера. Это не облегчает вычисление отображаемой ширины строки, за исключением ограниченного числа случаев, так как даже символ «фиксированной ширины» может быть получен комбинированием обычного символа с модифицирующим, который не имеет ширины. Например, буква «й» может быть получена из буквы «и» и [диакритического знака](https://ru.wikipedia.org/wiki/%D0%94%D0%B8%D0%B0%D0%BA%D1%80%D0%B8%D1%82%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B8%D0%B5_%D0%B7%D0%BD%D0%B0%D0%BA%D0%B8) «[крючок над буквой](https://ru.wikipedia.org/wiki/%D0%9A%D1%80%D0%B0%D1%82%D0%BA%D0%B0)». Сочетание таких знаков означает, что текстовые редакторы не могут рассматривать 32-битный код как единицу редактирования. Редакторы, которые ограничиваются работой с языками с письмом слева направо и [составными символами](https://ru.wikipedia.org/wiki/TrueType#.D0.A1.D0.BE.D1.81.D1.82.D0.B0.D0.B2.D0.BD.D0.BE.D0.B9_.D1.81.D0.B8.D0.BC.D0.B2.D0.BE.D0.BB) могут использовать символы фиксированного размера. Но такие редакторы вряд ли поддержат символы, лежащие за пределами нулевой (базовой) плоскости кодового пространства и вряд ли смогут работать одинаково хорошо с символами UTF-16.

* 1. **Привести примеры записей кодов символов**

**Utf-8**

Давайте возьмем символ «o» (англ). В таблице ASCII символов он находится на 111 позиции, в битовом виде это будет 01101111. В таблице юникода этот символ — U+006F что в битовом виде тоже будет 01101111.



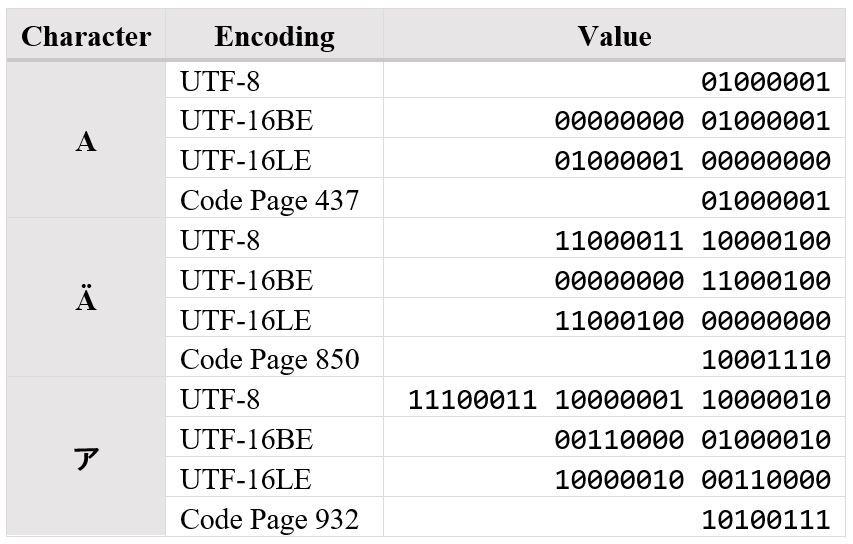
**Utf-16**

*символ «o» (латиница)* ***— 00000000 01101111***

*символ «M» (кириллица)* ***— 00000100 00011100***

**Сравнительная таблица**





* 1. **Разобраться в понятиях *code point*, *code unit*, *character set*, *coded character set*.**

**Кодовая точка - Code point** – числовое значение, представляющее символ в наборе кодированных символов

В терминологии кодировки символов, кодовая точка или позиция кода представляет собой любое числовое значение, составляющее пространство кода. Многие кодовые точки представляют собой отдельные символы, но они также могут иметь другие значения, например, для форматирования.

**Code point** ***–*** это примерно то же, что мы привыкли называть символом. Но не совсем. Например, буква «ё» может быть как одним codepoint'ом, так и двумя – буквой «е» и символом "две точки над предыдущей буквой"

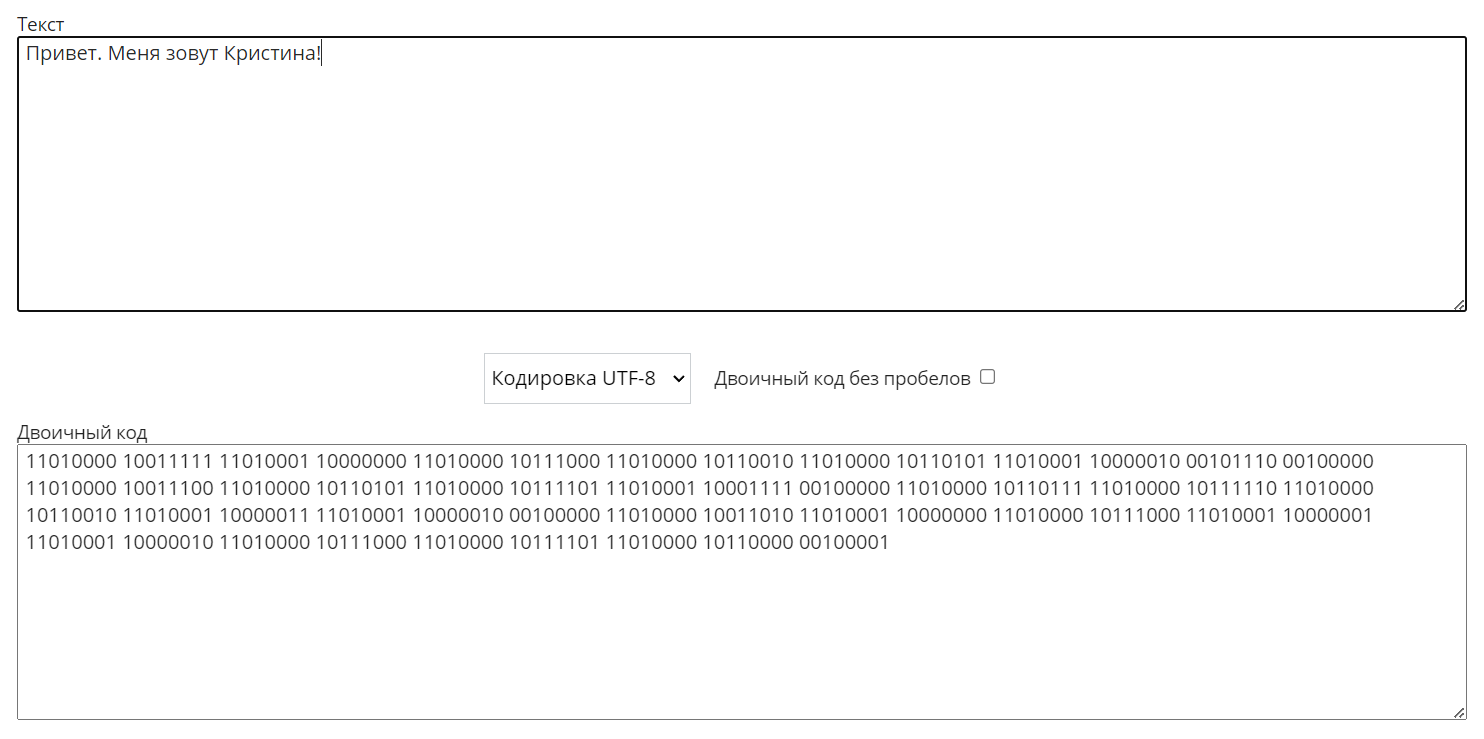
**Code unit** ***–*** это единицы кодировки. Байт для utf-8, Слово (два байта) для unf-16 или Длинное слово (четыре байта) для utf-32.

**Набо́р си́мволов (англ. character set)** – таблица, задающая кодировку конечного множества символов алфавита (обычно элементов текста: букв, цифр, знаков препинания). Такая таблица сопоставляет каждому символу последовательность длиной в один или несколько символов другого алфавита (точек и тире в коде Mорзе, сигнальных флагов на флоте, нулей и единиц (битов) в компьютере).

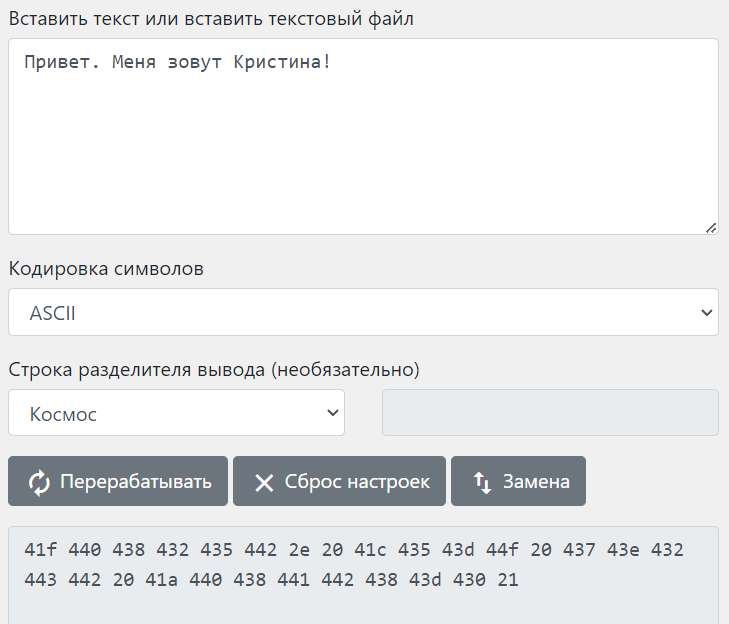
***Coded character set –*** набор кодов символов – содержит привязку каждого символа из репертуара к целому неотрицательному числу, называемому кодовой точкой (*code point*).

**UCS (англ. Universal Coded Character Set)** представляет собой стандартный набор символов, определенный международным стандартом ISO/IEC 10646, который является основой многих символьных кодировок. UCS содержит чуть более 128 000 абстрактных символов, как и в Unicode 9.0, каждый из которых определяется однозначно сочетанием имени и целого числа (так называемый "кодовый пункт"). Символы (буквы, цифры, идеограммы, логограммы и т.д.) из многих языков, манускриптов, традиций мира представлены в UCS с уникальными кодовыми пунктами. Символы из ранее не представленных письменностей добавляются в UCS, постоянно совершенствуя ее.

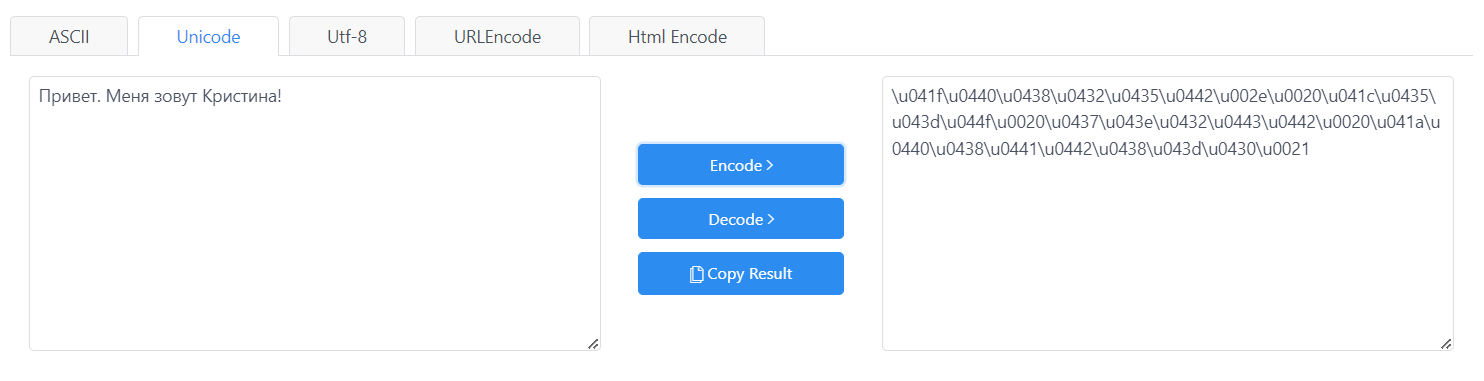
* 1. **Привести примеры кодирования текстовой информации.**



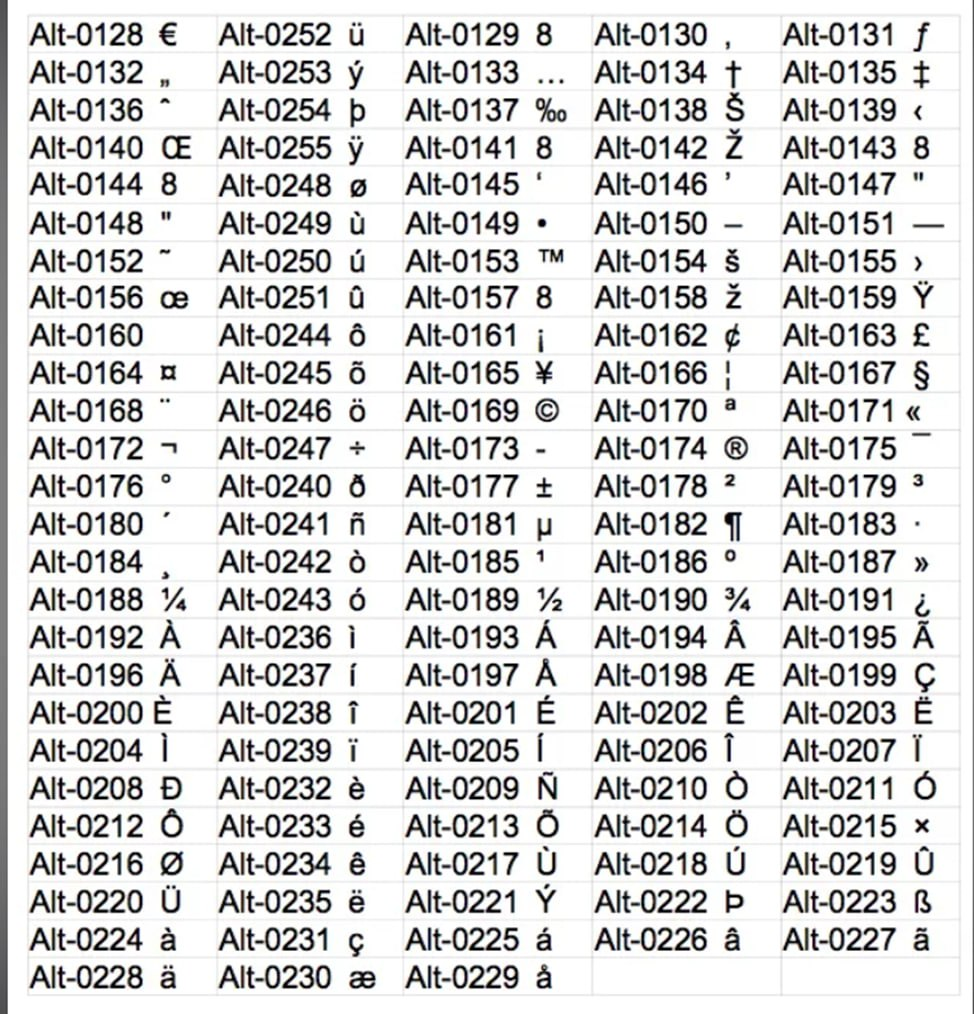
<https://allcalc.ru/node/1977>



<https://www.rapidtables.org/ru/convert/number/ascii-to-hex.html>



<https://checkserp.com/encode/unicode/>



**Выводы:** в данной лабораторной работе были рассмотрены основные стандарты кодирования информации: ASCII, ANSI и Unicode. Освоено понятие *Unicode* кодеки: *UTF*-8, *UTF*-16, *UTF*-32, правила записей кодов символов. Разобраны понятия *code point*, *code unit*, *character set*, *coded character set*.

Дополнительный вопрос: данный символ Ö набирался с помощью клавиш Alt-0214.