Лекция 02
БГТУ, ФИТ, 1 семестр

Основы программной инженерии (ПОИТ)

Системы программирования

План лекции:
- понятие программного обеспечения;
- системы программирования;
- данные, представление данных, кодировки;
- кодировка ASCII;
- стандарт кодирования Unicode;
- прямой (LE) и обратный (BE) порядок байт;
- маркер последовательности байтов (BOM).

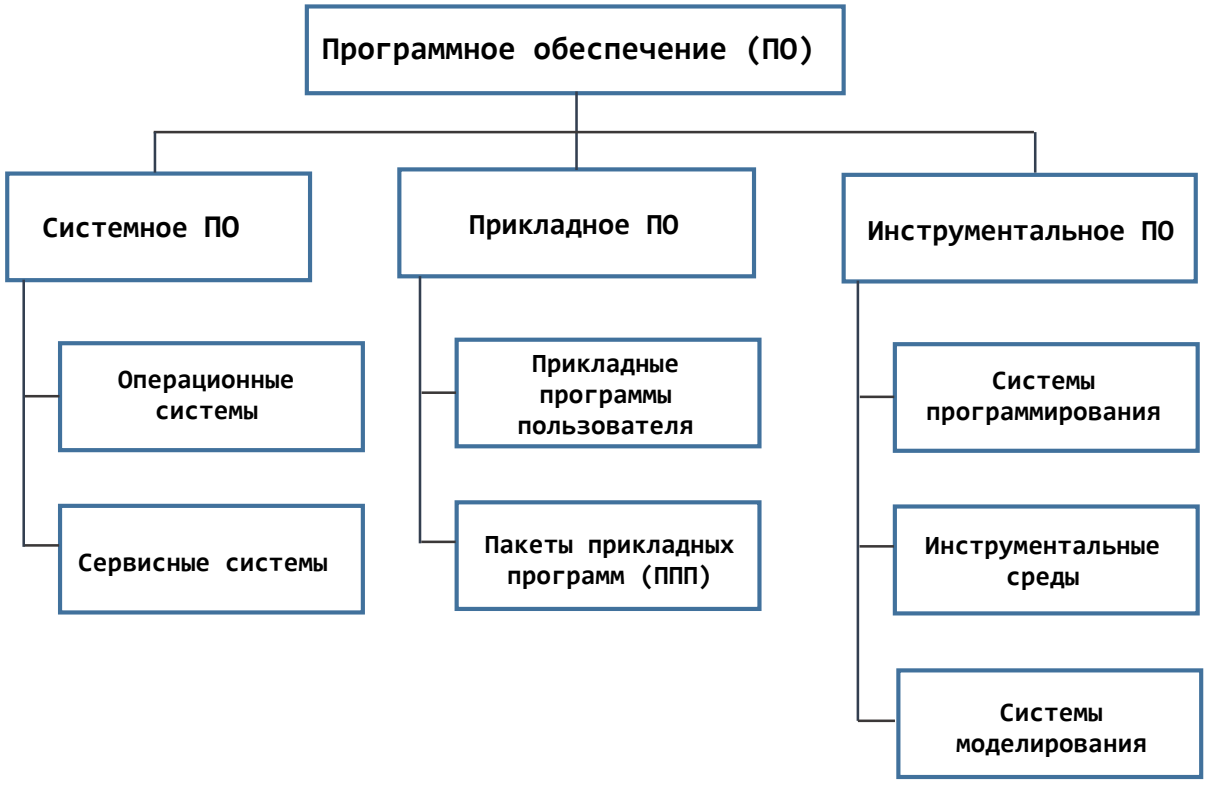
1. Программное обеспечение компьютера

Определение (ГОСТ Р 51904-2002)

> **Программное обеспечение (ПО)** – совокупность компьютерных программ и программных документов, необходимых для эксплуатации этих программ.

Определение (ISO/IEC 26514:2008)

> **Программное обеспечение (ПО)** – программа или множество программ, используемых для управления компьютером.



Классификация программного обеспечения:

Системное ПО – комплекс программ, которые обеспечивают управление компонентами компьютерной системы:

- управление ресурсами компьютера;
- создание копий используемой информации;
- проверка работоспособности устройств компьютера;
- и др.

Прикладное ПО – предназначено для выполнения определённых пользовательских задач и рассчитанная на непосредственное взаимодействие с пользователем.

Инструментальное ПО служит для автоматизации процесса разработки.

Операционная система – комплекс системных программ, расширяющий возможности вычислительной системы, обеспечивающий управление её ресурсами, загрузку и выполнение прикладных программ, взаимодействие с пользователями.

Системы программирования – системные программы, предназначенные для разработки программного обеспечения.

2. Система программирования

Система программирования:

комплекс программных средств, предназначенных для автоматизации процесса разработки, отладки программного обеспечения и подготовки программного кода к выполнению

Новые требования (тенденции) в современной технологии разработке программного обеспечения:

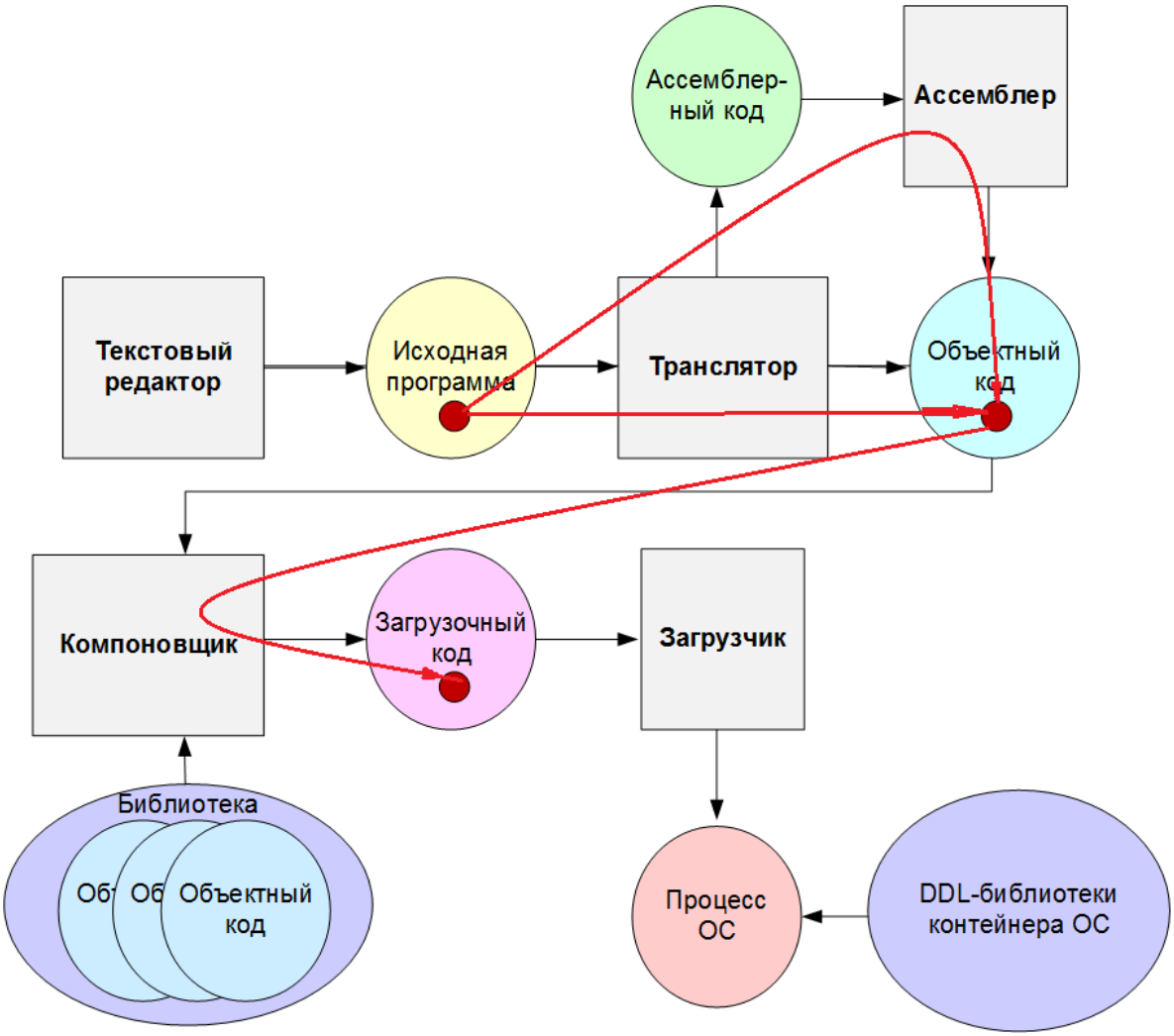
- распространение промышленных методов организации (планирование трудозатрат, учет, контроль результатов, и т.п.) при проведении работ по работ по разработке программного обеспечения:
- перенос акцента с процесса программирования на более ранние стадии – анализ предметной области, формирование требований.

3. Состав системы программирования:

Состав системы программирования	трансляторы компоновщики отладчики профилировщики программные библиотеки редакторы кода системы поддержки версий и пр.

Система программирования является основным инструментом программиста.

4. Структура классической системы программирования



От исходного кода к исполняемому модулю, основные этапы преобразования:

Классическая схема создания исполняемого файла выполняется для компилируемых языков:

(1) обработка исходного кода препроцессором,

(2) компиляция в объектный код и

(3) компоновка объектных модулей, включая модули из объектных библиотек, в исполняемый файл.

5. Язык программирования:

Язык программирования	формальная знаковая система, предназначенная для записи компьютерных программ. Знаковая система определяет набор лексических, синтаксических и семантических правил написания программы (программного кода). Язык программирования представляется в виде набора спецификаций, определяющих его синтаксис и семантику.

Язык программирования представляется в виде набора спецификаций, определяющих его синтаксис и семантику.
Язык программирования определяется не только через спецификации стандарта языка, формально определяющие его синтаксис и семантику, но и через реализации стандарта (программные средства, обеспечивающих трансляцию или интерпретацию программ на этом языке), которые различаются по производителю, версии, времени выпуска, полноте воплощения стандарта, дополнительным возможностям; могут иметь определённые ошибки или особенности реализации.
Спецификация системы программирования: набор требований к системе программирования, достаточный для ее разработки.

6. Кодирование информации

Текстовая информация выражается с помощью естественных или формальных языков в письменной или печатной форме.

Пример:

преподаватель читает лекцию → процесс кодирования → *студент* делает для себя пометки → процесс декодирования → *студент* использует конспект

Системы счисления

десятичная	двоичная	шестнадц.
0	0000	0
1	0001	1
2	0010	2
...	...	...
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

Степень двойки

2^0	1
2^1	2
2^2	4
2^3	8
2^4	16
2^5	32
2^6	64
2^7	128
2^8	256

Данные в памяти компьютера

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
0	0	0	0	1	0	0	0

старший бит — младший бит

байт – минимальная адресуемая единица
бит – минимальная единица хранения (0 или 1)

Решение проблем с кодировкой текста	Текстовый символ кодируется его порядковым номером (0-127), представленным в двоичной системе счисления (1963 ASCII).

Ранние языки, возникшие в эпоху 6-битных символов, использовали более ограниченный набор.

> ***Пример***:
> алфавит Фортрана включает 49 символов: A B C D E F G H I J K L M N O P Q R S T U V W X Y Z 0 1 2 3 4 5 6 7 8 9 = + - * / () . , $ ' : пробел

a. Американский стандартный код для обмена информацией. ASCII

ASCII (American Standard Code for Information Interchange) – американский стандартный код для обмена информацией.
ASCII – 8-битная кодировка для представления десятичных цифр, латинского и национального алфавитов, знаков препинания и управляющих символов.

Таблица кодов ASCII делится на две части:
Международным стандартом является первая половина таблицы, т.е. символы с номерами от 0 (00000000), до 127 (01111111).
К концу 1980-х годов стандартом стали 8-битные кодировки.

ASCII Code Chart

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2		!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	DEL

***Расширенные таблицы*:** в расширенных таблицах символы с порядковыми номерами 128-255 представляют символы национальных языков.

Переносимый набор символов	является базовым алфавитом для практически всех современных языков программирования.

Переносимый набор символов (portable character set) – набор из 103 символов, которые должны присутствовать в любой используемой кодировке (стандарт POSIX).

POSIX (англ. Portable Operating System Interface – переносимый интерфейс операционных систем) – набор стандартов, описывающих интерфейсы между операционной системой (ОС) и прикладной программой (системный API), библиотеку языка C и набор приложений и их интерфейсов.

Переносимый набор символов включает в себя все печатные символы US-ASCII и часть управляющих и является базовым алфавитом для практически всех современных языков программирования.

- Альтернативная кодировка **CP866** (операционная система MS-DOS):

Все специфические европейские символы во второй половине таблицы CP866 заменены на кириллицу, а псевдографические символы оставлены без изменения.

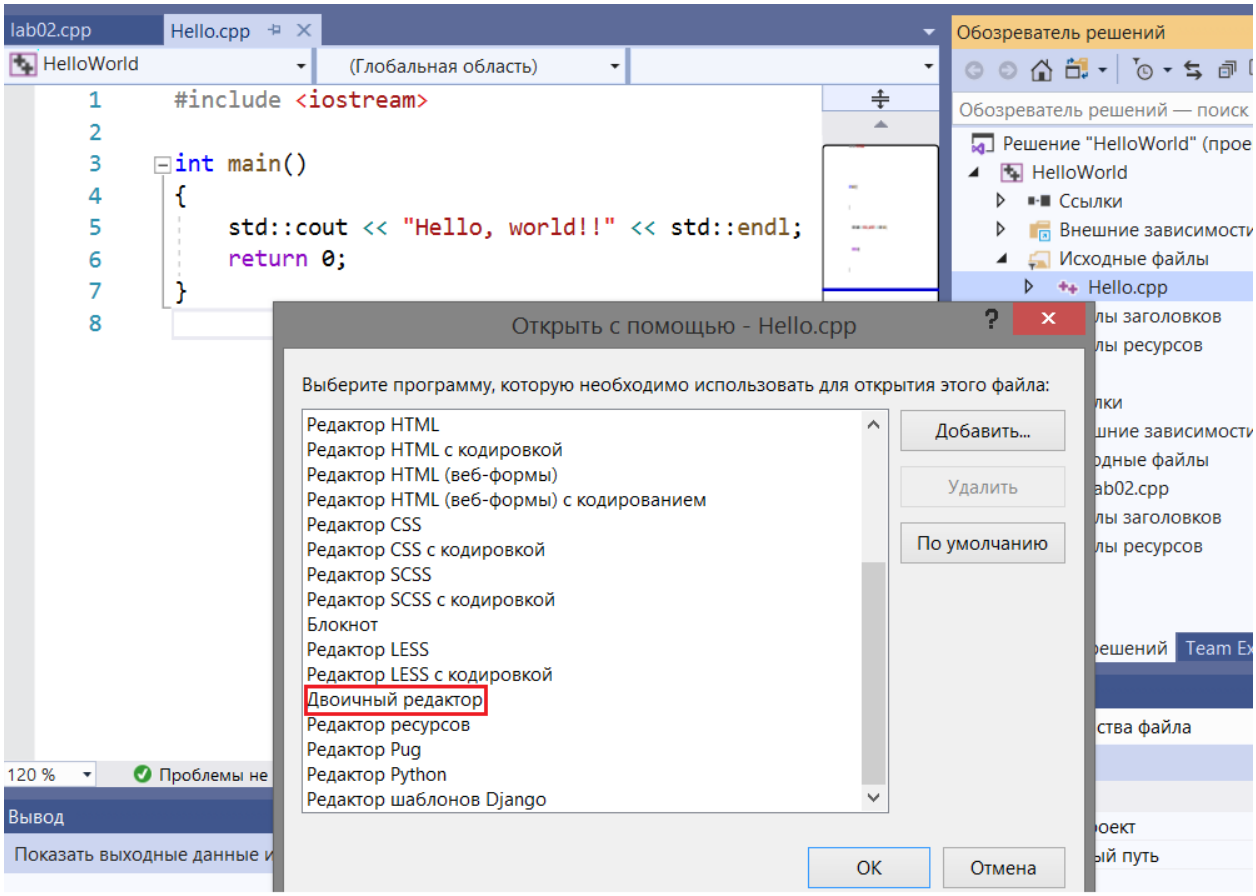
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
8	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
9	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я
A	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
B	░	▒	▓	│	┤	╡	╢	╖	╕	╣	║	╗	╝	╜	╛	┐
C	└	┴	┬	├	─	┼	╞	╟	╚	╔	╩	╦	╠	═	╬	╧
D	╨	╤	╥	╙	╘	╒	╓	╫	╪	┘	┌	▉	▄	▌	▐	▀
E	р	с	т	у	ф	х	ц	ч	ш	щ	ъ	ы	ь	э	ю	я
F	Ё	ё	Є	є	Ї	ї	Ў	ў	°	∙	·	√	№	¤	■	

- русская Windows-кодировка (**Windows-1251**, синоним **CP1251**)

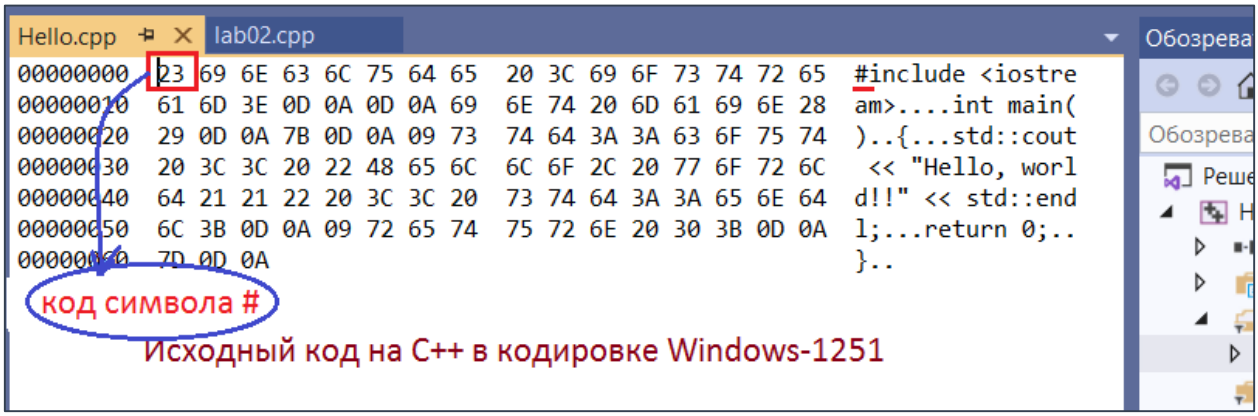
Windows-1251 – набор символов и кодировка, являющаяся стандартной 8-битной кодировкой для русских версий Microsoft Windows до 10-й версии.

	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
00	NUL 0000	STX 0001	SOT 0002	ETX 0003	EOT 0004	ENQ 0005	ACK 0006	BEL 0007	BS 0008	HT 0009	LF 000A	VT 000B	FF 000C	CR 000D	SO 000E	SI 000F
10	DLE 0010	DC1 0011	DC2 0012	DC3 0013	DC4 0014	NAK 0015	SYN 0016	ETB 0017	CAN 0018	EM 0019	SUB 001A	ESC 001B	FS 001C	GS 001D	RS 001E	US 001F
20	SP 0020	! 0021	" 0022	# 0023	$ 0024	% 0025	& 0026	' 0027	(0028	) 0029	* 002A	+ 002B	, 002C	- 002D	. 002E	/ 002F
30	0 0030	1 0031	2 0032	3 0033	4 0034	5 0035	6 0036	7 0037	8 0038	9 0039	: 003A	; 003B	< 003C	= 003D	> 003E	? 003F
40	@ 0040	A 0041	B 0042	C 0043	D 0044	E 0045	F 0046	G 0047	H 0048	I 0049	J 004A	K 004B	L 004C	M 004D	N 004E	O 004F
50	P 0050	Q 0051	R 0052	S 0053	T 0054	U 0055	V 0056	W 0057	X 0058	Y 0059	Z 005A	[005B	\ 005C	] 005D	^ 005E	_ 005F
60	` 0060	a 0061	b 0062	c 0063	d 0064	e 0065	f 0066	g 0067	h 0068	i 0069	j 006A	k 006B	l 006C	m 006D	n 006E	o 006F
70	p 0070	q 0071	r 0072	s 0073	t 0074	u 0075	v 0076	w 0077	x 0078	y 0079	z 007A	{ 007B	| 007C	} 007D	~ 007E	DEL 007F
80	Ђ 0402	Ѓ 0403	‚ 201A	ѓ 0453	„ 201E	… 2026	† 2020	‡ 2021	€ 20AC	‰ 2030	Љ 0409	‹ 2039	Њ 040A	Ќ 040C	Ћ 040B	Џ 040F
90	ђ 0452	‘ 2018	’ 2019	“ 201C	” 201D	• 2022	– 2013	— 2014		™ 2122	љ 0459	› 203A	њ 045A	ќ 045C	ћ 045B	џ 045F
A0	NBSP 00A0	Ў 040E	ў 045E	Ј 0408	¤ 00A4	Ґ 0490	¦ 00A6	§ 00A7	Ё 0401	© 00A9	Є 0404	« 00AB	¬ 00AC	- 00AD	® 00AE	Ї 0407
B0	° 00B0	± 00B1	І 0406	і 0456	ґ 0491	µ 00B5	¶ 00B6	· 00B7	ё 0451	№ 2116	є 0454	» 00BB	ј 0458	Ѕ 0405	ѕ 0455	ї 0457
C0	А 0410	Б 0411	В 0412	Г 0413	Д 0414	Е 0415	Ж 0416	З 0417	И 0418	Й 0419	К 041A	Л 041B	М 041C	Н 041D	О 041E	П 041F
D0	Р 0420	С 0421	Т 0422	У 0423	Ф 0424	Х 0425	Ц 0426	Ч 0427	Ш 0428	Щ 0429	Ъ 042A	Ы 042B	Ь 042C	Э 042D	Ю 042E	Я 042F
E0	а 0430	б 0431	в 0432	г 0433	д 0434	е 0435	ж 0436	з 0437	и 0438	й 0439	к 043A	л 043B	м 043C	н 043D	о 043E	п 043F
F0	р 0440	с 0441	т 0442	у 0443	ф 0444	х 0445	ц 0446	ч 0447	ш 0448	щ 0449	ъ 044A	ы 044B	ь 044C	э 044D	ю 044E	я 044F

Интегрированная среда разработки Visual Studio. Открытие файла в двоичном редакторе:



Представление в памяти файла с исходным кодом:



Представление символьной информации в кодировке Windows-1251:

lab02.cpp

```
00000000  23 69 6E 63 6C 75 64 65  20 3C 69 6F 73 74 72 65  #include <iostre
00000010  61 6D 3E 0D 0A 0D 0A 69  6E 74 20 6D 61 69 6E 28  am>....int main(
00000020  29 0D 0A 7B 0D 0A 09 63  68 61 72 20 68 65 6C 6C  )..{...char hell
00000030  6F 5B 5D 20 3D 20 22 48  65 6C 6C 6F 2C 20 22 3B  o[] = "Hello, ";
00000040  0D 0A 09 63 68 61 72 20  66 69 6F 5B 5D 20 3D 20  ...char fio[] =
00000050  22 49 76 61 6E 6F 76 20  49 76 61 6E 20 49 76 61  "Ivanov Ivan Iva
00000060  6E 6F 76 69 63 68 22 3B  0D 0A 09 73 74 64 3A 3A  novich";...std::
00000070  63 6F 75 74 20 3C 3C 20  68 65 6C 6C 6F 20 3C 3C  cout << hello <<
00000080  20 66 69 6F 20 3C 3C 20  73 74 64 3A 3A 65 6E 64   fio << std::end
00000090  6C 3B 0D 0A 09 72 65 74  75 72 6E 20 30 3B 0D 0A  l;...return 0;..
000000a0  7D 0D 0A                                          }..
```

Исходный код lab02.cpp

b. Международный стандарт UNICODE

Решение проблем	неправильного декодирования; ограниченность набора символов; преобразования из одной кодировки в другую; проблема дублирования шрифтов.

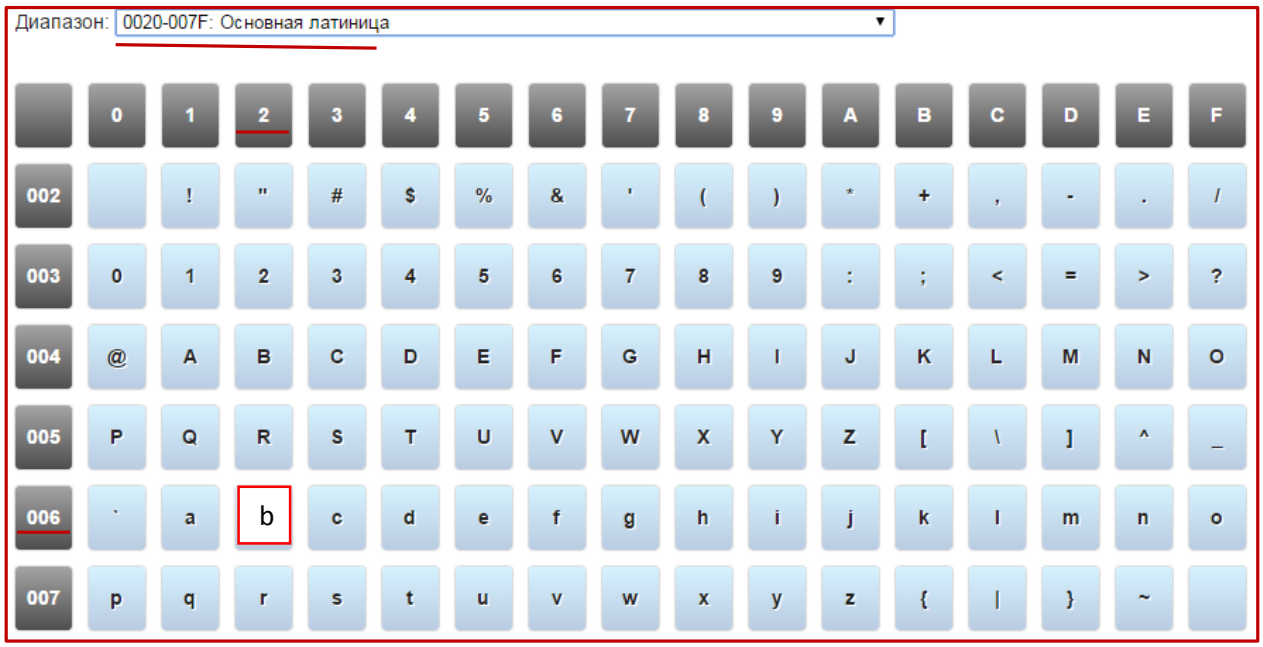
Стандарт предложен в 1991 году некоммерческой организацией Unicode Consortium, стандарт ISO/IEC 10646:2020.

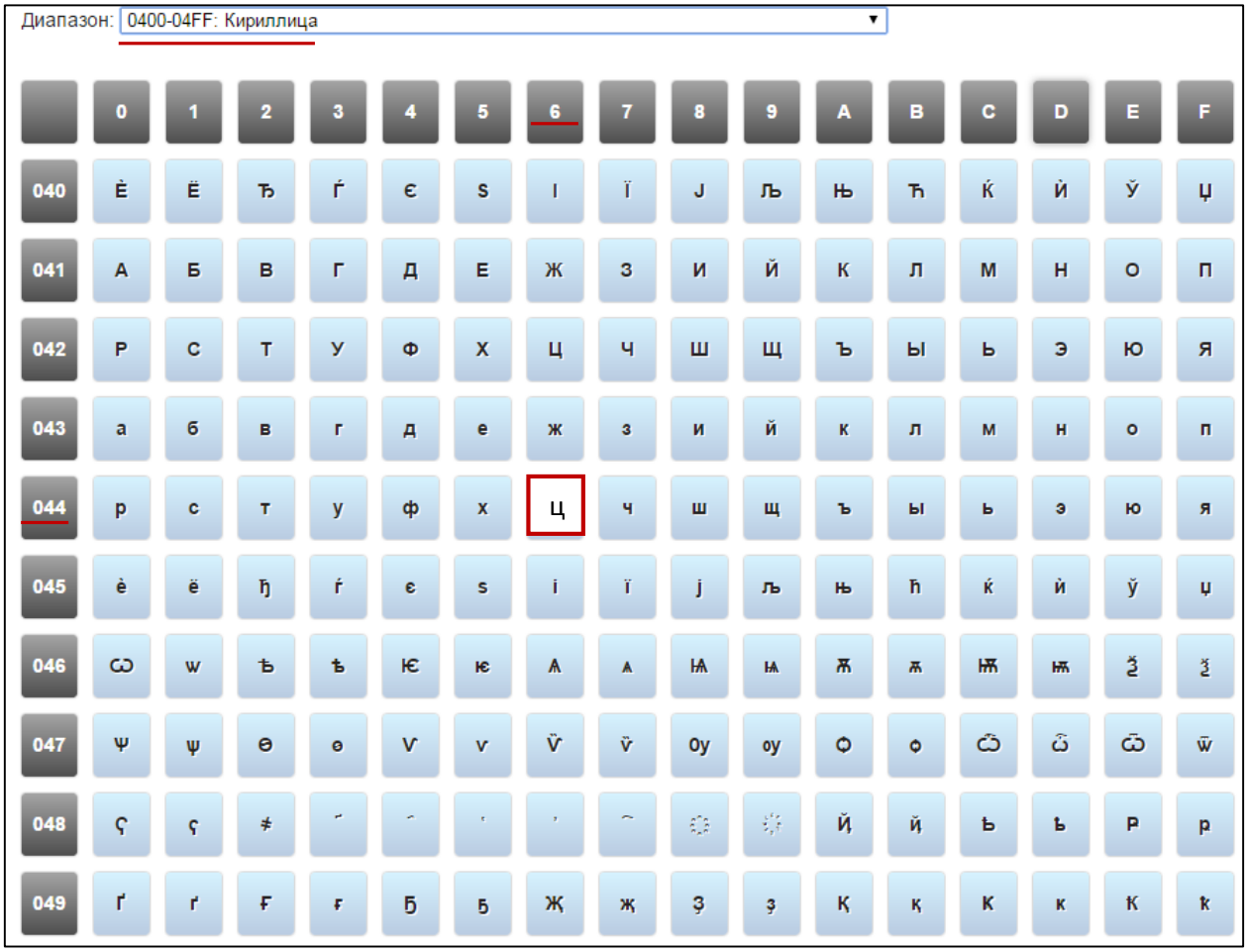
UNICODE	**Юникод** – стандарт кодирования символов, позволяющий представить знаки почти всех письменных языков, состоит из 2х разделов: – **UCS** – universal character set (универсальный набор символов); – **UTF** – Unicode transformation format (семейство кодировок).

Принято обозначение символа **U+xxx**, где **xxx**- число в шестнадцатеричном формате.

- **UNICODE:**
  - UCS расположены в 17 плоскостях (0-16);
  - в каждой плоскости 2^{16} (65 536) символов;
  - плоскость 0 – основная (основные символы);
  - 1-14 – дополнительные;
  - 15-16 – для частного использования.

- **UNICODE: http://foxtools.ru/Unicode**





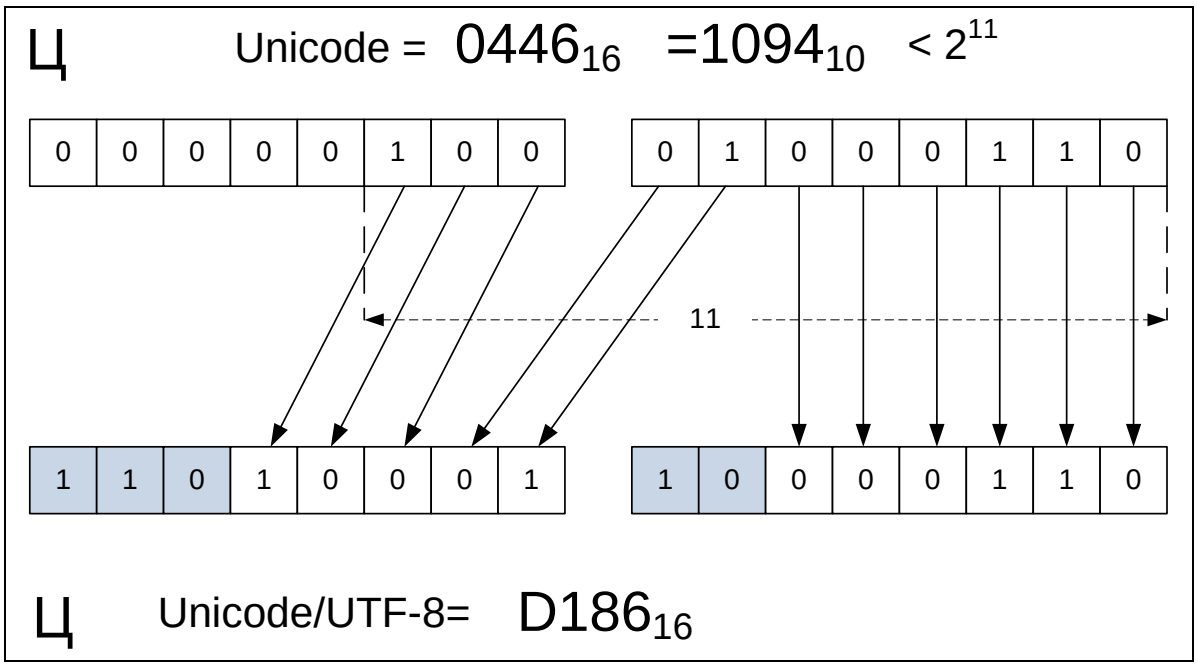
- **UNICODE:** ***кодировка UTF-8***

UTF-8 — представление Юникода, обеспечивающее совместимость со старыми системами, использовавшими 8-битные символы.

Алгоритм кодирования в UTF-8:

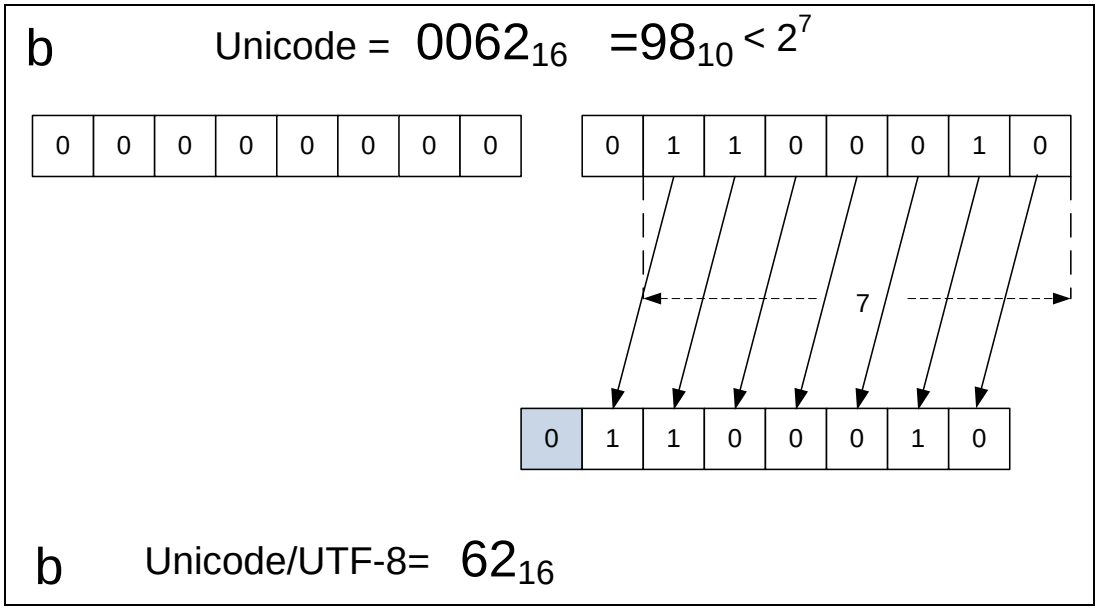
1) определить количество октетов (октет: 8 битов или 1 байт) – т.е. в какой диапазон значений попадает количество значащих символов (7, 11, 16, 21, 26, 31);
2) подготовить старшие биты первого октета:
   a. 0xxxxxxx для ***одного*** октета;
   b. 110xxxxx – для ***двух***;
   c. 1110xxxx – для ***трех*** и т.д..
   d. 10xxxxxx – для ***остальных*** октетов;
3) заполнить оставшиеся биты (выше обозначены как x) в октетах кодом символа Юникода в двоичном виде. Начать с младших битов, поставив их в младшие биты последнего октета кода. И так далее, пока все биты кода символа не будут перенесены в свободные биты октетов.

Пример:



$0446_{16} = 4*16^2 + 4*16 + 6 = 1094_{10}$

Пример:



UNICODE: ***кодировка UTF-8.*** Для символов в диапазоне:
0x00000000 ÷ 0x0000007F: 0xxxxxxx (один октет)
0x00000080 ÷ 0x000007FF: 110xxxxx 10xxxxxx (два октета)
0x00000800 ÷ 0x0000FFFF: 1110xxxx 10xxxxxx 10xxxxxx (три октета)
0x00010000 ÷ 0x001FFFFF: 11110xxx 10xxxxxx 10xxxxxx 10xxxxxx

UNICODE: ***кодировка UTF-16***
В UTF-16 символы кодируются двухбайтовыми словами (16 битов) с использованием всех возможных диапазонов значений (от 0 до FFFF16).

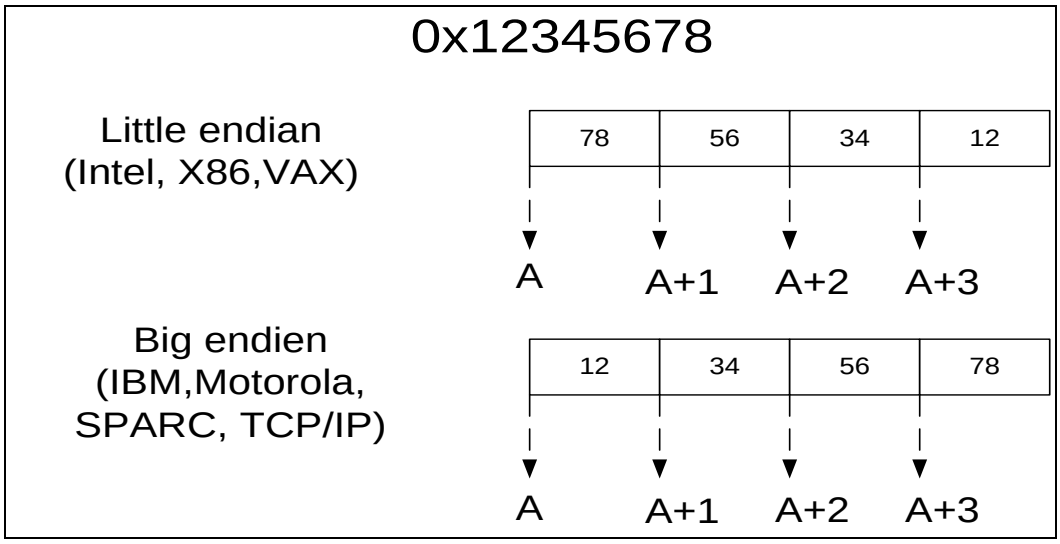
- **Маркер последовательности байтов UNICODE: BOM** (Byte Order Mark)

Для определения формата представления Юникода в начало текстового файла записывается сигнатура (обозначение) — символ U+FEFF — маркер последовательности байтов.

Шестнадцатеричное представление маркера последовательности байтов для кодировок:

Кодировка	Представление (hex)
UTF-8	EF BB BF
UTF-16 (BE)	FE FF
UTF-16 (LE)	FF FE
UTF-32 (BE)	00 00 FE FF
UTF-32 (LE)	FF FE 00 00

- **Порядок следования байтов:**
  - **LE** (Little endian order, прямой порядок, от младшего к старшему);
  - **BE** (Big endian order, обратный порядок, от старшего к младшему).



Представление в памяти целочисленного числа на платформе x86:
порядок следования байтов LE
(Little endian order, прямой порядок, от младшего к старшему)