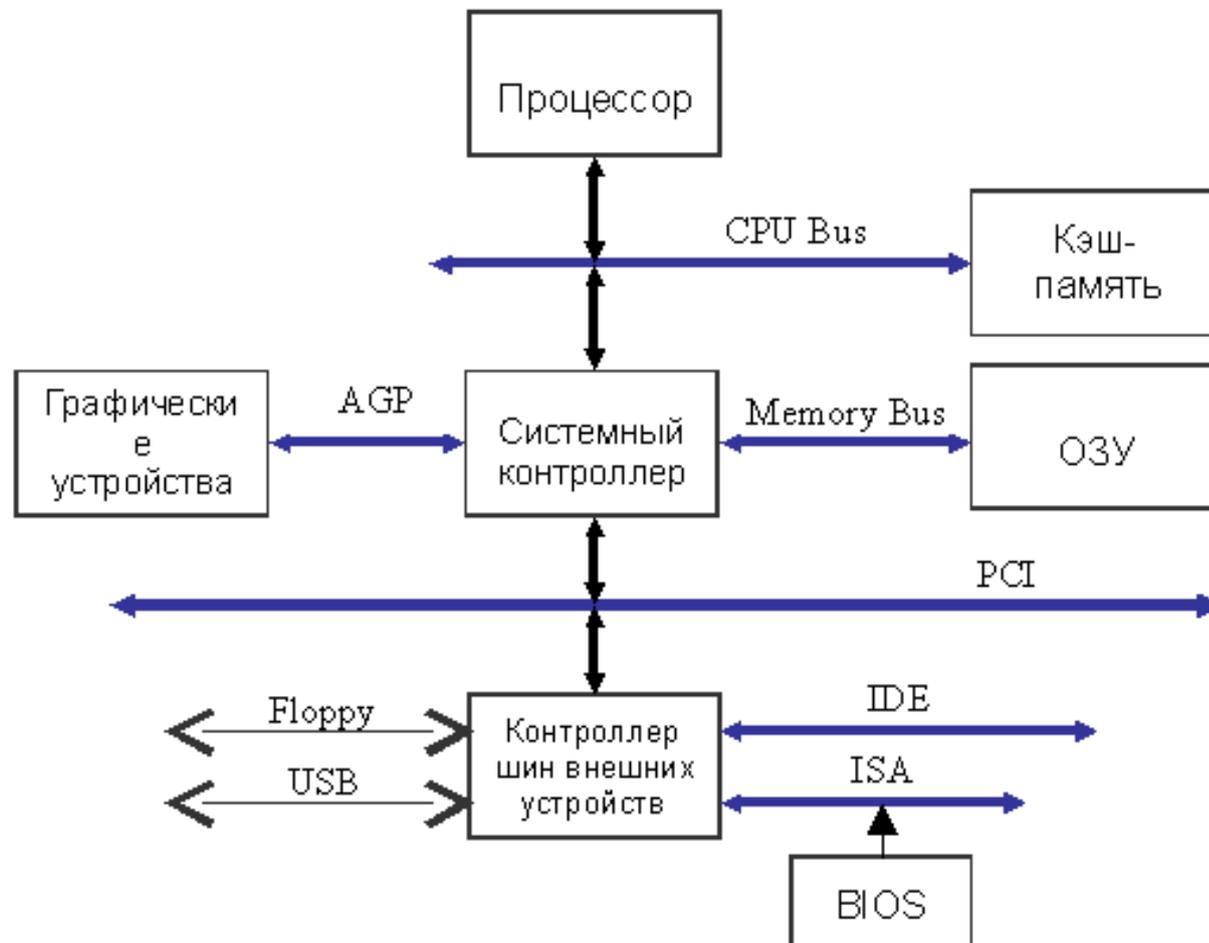


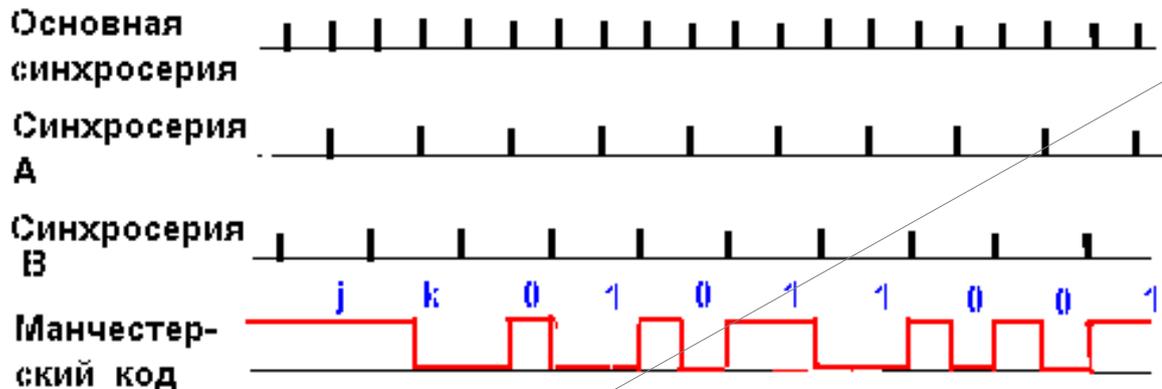
# Лекция №2

Способы представления и кодирования информации

# Архитектура ЭВМ



# Представление информации



210 229 245



Технология обработки информации

1101001011110010111110101111011011110111011  
 101011111011101110001111101000111111110010  
 00001110111011100001111100001110000011100  
 00111101110111100101110101011101000001000  
 00111010001110110111110100111011101111000  
 01110110011100000111101101110100011101000  
 00001010.....

# ASCII

ASCII (англ. American Standard Code for Information Interchange) — американская стандартная кодировочная таблица для печатных символов и некоторых специальных кодов.

ASCII представляет собой кодировку для представления десятичных цифр, латинского и национального алфавитов, знаков препинания и управляющих символов. Изначально разработанная как 7-битная, с широким распространением 8-битного байта ASCII стала восприниматься как половина 8-битной. В компьютерах обычно используют расширения ASCII с задействованной второй половиной байта.

# ASCII (7bit)

	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	SPC	!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/
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

# ASCII (8bit)

128	Ç	144	É	161	í	177	⌘	193	⊥	209	≠	225	β	241	±
129	ü	145	æ	162	ó	178	⌘	194	⊤	210	π	226	Γ	242	≥
130	é	146	Æ	163	ú	179		195	⊥	211	⊥	227	π	243	≤
131	â	147	ô	164	ñ	180	⊥	196	—	212	⊥	228	Σ	244	∫
132	ä	148	ö	165	Ñ	181	⊥	197	⊥	213	≠	229	σ	245	∫
133	à	149	ò	166	ª	182	⊥	198	⊥	214	≠	230	μ	246	÷
134	â	150	û	167	º	183	π	199	⊥	215	⊥	231	τ	247	≈
135	ç	151	ù	168	¿	184	⊥	200	⊥	216	≠	232	Φ	248	°
136	ê	152	—	169	—	185	⊥	201	≠	217	⊥	233	⊗	249	.
137	ë	153	Ö	170	¬	186	⊥	202	≠	218	⊥	234	Ω	250	.
138	è	154	Ü	171	½	187	⊥	203	≠	219	■	235	δ	251	√
139	ï	156	£	172	¼	188	⊥	204	⊥	220	■	236	∞	252	—
140	î	157	¥	173	¡	189	⊥	205	=	221	■	237	φ	253	z
141	ì	158	—	174	«	190	⊥	206	≠	222	■	238	ε	254	■
142	Ä	159	ƒ	175	»	191	⊥	207	≠	223	■	239	∩	255	
143	Å	160	á	176	⌘	192	⊥	208	⊥	224	α	240	≡		

# ASCII (8bit, OEM, DOS, 866)

	00	01	02	03	04	05	06	07	08	09	0a	0b	0c	0d	0e	0f
00		␣	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣
10	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣
20		!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/
30	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
60	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~
80	А	Б	В	Г	Д	Е	Ж	Ш	И	Й	К	Л	М	Н	О	П
90	Р	С	Т	У	Ф	Х	Ц	Ч	с	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	Л	Л	Т	Т	—	†	†			л	л			=		†
d0	л	л	т	т	—	†	†			л	л			=		†
e0	р	р	с	с	ф	х	ц	ч	ш	щ	ъ	ы	ь	э	ю	я
f0	Р	Р	С	С	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
0				©	È	€	€	.	.	°							
1		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
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	{		}	~	□
8		Ъ	Г	,	ѓ	„	…	†	‡		‰	Љ	«	Њ	К	Ћ	Ц
9		ђ	'	'	"	"	—	—		™	љ	›	њ	ќ	ћ	џ	
A		У	Ў	Ј	Ѡ	Ґ		§	È	©	€	«		-	®	İ	
B		°	±	І	і	ґ	μ	.	ё	№	€	»	ј	Ѕ	ѕ	ї	
C		А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
D		Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я
E		а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
F		р	с	т	у	ф	х	ц	ч	ш	щ	ъ	ы	ь	э	ю	я

# Кодировки с поддержкой русского языка

1. 866 (CP866)
2. 1251 (CP1251, WINDOWS-1251)
3. KOI8-R
4. ISO-8859-5
5. UTF-8

# UTF

Юнико́д или Унико́д (англ. Unicode) — стандарт кодирования символов, позволяющий представить знаки практически всех письменных языков.

Стандарт предложен в 1991 году некоммерческой организацией Unicode Consortium. Применение этого стандарта позволяет закодировать очень большое число символов из разных письменностей: в документах Unicode могут соседствовать китайские иероглифы, математические символы, буквы греческого алфавита, латиницы и кириллицы, при этом становится ненужным переключение кодовых страниц.

# UTF

Стандарт состоит из двух основных разделов: универсальный набор символов (англ. UCS, universal character set) и семейство кодировок (англ. UTF, Unicode transformation format).

Коды в стандарте Юникод разделены на несколько областей. Область с кодами от U+0000 до U+007F содержит символы набора ASCII с соответствующими кодами. Далее расположены области знаков различных письменностей, знаки пунктуации и технические символы. 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

# UTF

Первая версия Юникода представляла собой кодировку с фиксированным размером символа в 16 бит, то есть общее число кодов было  $2^{16}$  (65 536). Отсюда происходит практика обозначения символов четырьмя шестнадцатеричными цифрами (например, U+0410). Версии:

1.1 (ISO/IEC 10646—1:1993), стандарт 1991—1995 годов.

2.0, 2.1 (ISO/IEC 10646—1:1993 + дополнения), стандарт 1996 года.

3.0 (стандарт ISO/IEC 10646—1:2000), стандарт 2000 года.

3.2, стандарт 2002 года.

4.0, стандарт 2003.

4.01, стандарт 2004.

4.1, стандарт 2005.

5.0, стандарт 2006.

5.1, стандарт 2008.

5.2, стандарт 2009.

6.0, стандарт 2010.

6.1, стандарт 2012.

# UTF-16

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.

UTF-16 была создана для совместимости со старыми 16-битными, где первые 65 536 позиций, за исключением позиций из интервала U+D800...U+DFFF, отображаются непосредственно как 16-битные числа, а остальные представляются в виде «суррогатных пар» (первый элемент пары - U+D800...U+DBFF, второй элемент - U+DC00...U+DFFF). Для суррогатных пар была использована часть кодового пространства (2048 позиций), ранее отведённого для «символов для частного использования». Поскольку в UTF-16 можно отобразить только  $2^{20} + 2^{16} - 2048$  (1 112 064) символов, то это число и было выбрано в качестве окончательной величины кодового пространства Юникода.

# UTF-8

UTF-8 — представление Юникода, обеспечивающее наилучшую совместимость со старыми системами, использовавшими 8-битные символы. Текст, состоящий только из символов с номером меньше 128, при записи в UTF-8 превращается в обычный текст ASCII. И наоборот, в тексте UTF-8 любой байт со значением меньше 128 изображает символ ASCII с тем же кодом. Остальные символы Юникода изображаются последовательностями длиной от 2 до 6 байт (на деле, только до 4 байт, поскольку в Юникоде нет символов с кодом больше 10FFFF, и вводить их в будущем не планируется), в которых первый байт всегда имеет вид 11xxxxxx, а остальные — 10xxxxxx.

# BOM

Для определения формата представления Юникода в начало текстового файла записывается сигнатура — символ U+FEFF (неразрывный пробел с нулевой шириной), также именуемый меткой порядка байтов (англ. byte order mark, BOM). Это позволяет различать UTF-кодировки.

UTF-8

EF BB BF

UTF-16BE

FE FF

UTF-16LE

FF FE

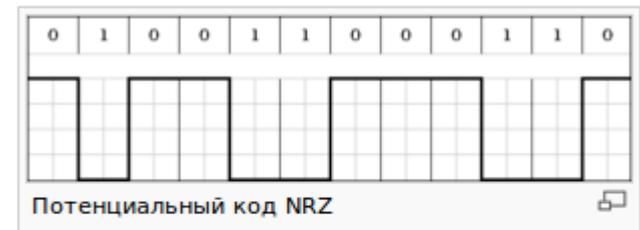
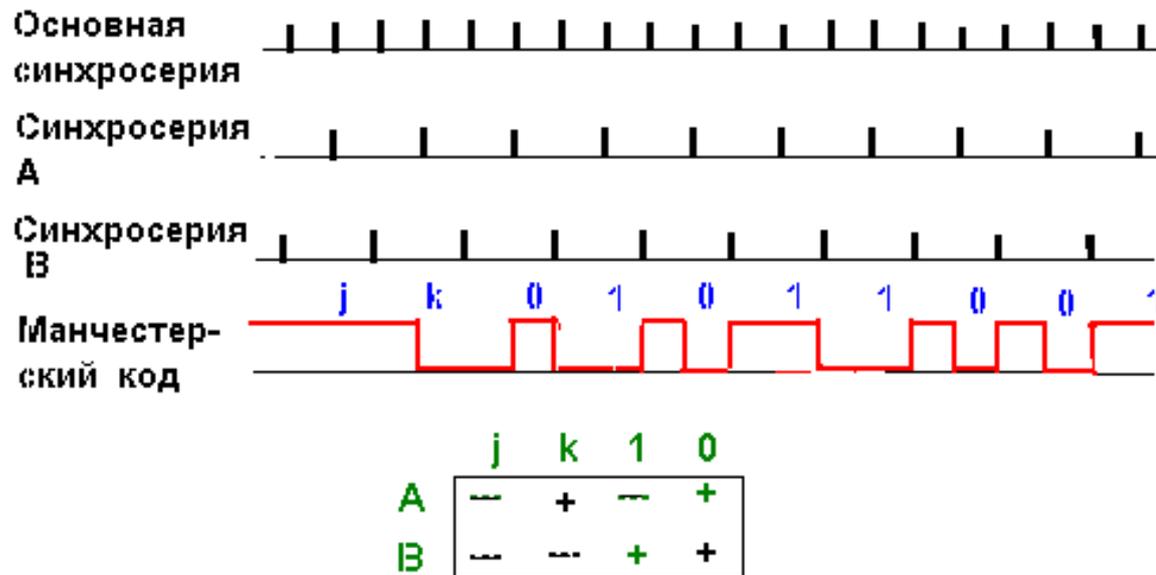
UTF-32BE

00 00 FE FF

UTF-32LE

FF FE 00 00

# Представление информации в сетях



Каждый бит кодового слова передается или записывается с помощью дискретных сигналов, например, импульсов. Способ представления исходного кода определёнными сигналами определяется форматом кода.

# Интерфейсы



Последовательные (COM)

Параллельные (LTP)

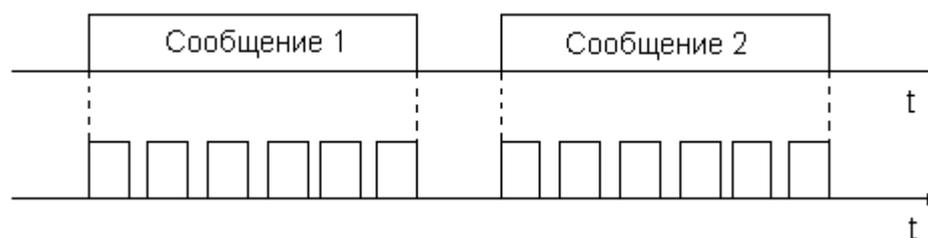


# Способы передачи

При обмене данными по каналам связи используются три метода передачи данных:

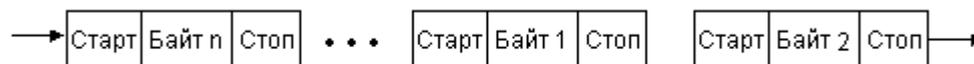
- 1) Симплексная (однонаправленная) — TV, радио;
- 2) Полудуплексная передача — (приём и передача данных осуществляются поочерёдно);
- 3) Дуплексная (двунаправленная) — каждая станция одновременно передаёт и принимает данные.

# Способы передачи



Синхронный

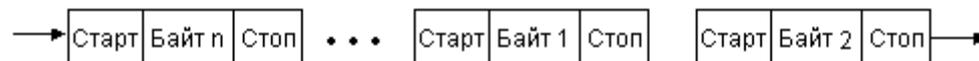
Асинхронный



# Передача данных

Данные преобразуются в последовательный формат, то есть в сплошную цепочку битов. Затем, чтобы данные можно было передавать асинхронно, эта цепочка разбивается на блоки, которые в данном случае равны одному байту. И наконец, каждый блок снабжается маркерами — признаками начала и конца блока.

Зачастую этими маркерами являются стартовый бит (start bit) и стоповый бит (stop bit). Стартовый бит равен 0, стоповый — 1. Послав один за другим стартовый бит, байт данных и стоповый бит, передатчик может повторять стоп-бит до тех пор, пока приемник не будет готов получить следующий байт. Тогда посылается стартовый бит, служащий сигналом начала новой порции данных.





# Передача данных

Такой механизм обеспечивает гибкий контакт передатчика и приемника, но не защищает от ошибок, которые могут произойти при передаче. Чтобы обеспечить обнаружение ошибок, стандарты порой предписывают вставку сразу после байта данных (перед стоповым битом) еще одного бита — контрольного (parity bit). Его значение определяется содержимым передаваемого байта и тем, какой из двух режимов контроля установлен — контроль четности (even parity) или контроль нечетности (odd parity).

Допустим, передается байт 01001101. Он содержит четыре, т. е. четное количество единиц. Поэтому, если включен режим контроля нечетности, контрольный бит устанавливается в 1, чтобы общее число единиц стало нечетным. И наоборот, когда установлен режим контроля четности, контрольный бит приравнивается 0, чтобы сохранить количество единиц четным.

# Передача данных



При синхронном режиме передачи пользовательские данные собираются в кадр, который предваряется байтами синхронизации. Старт-стопные биты между соседними байтами отсутствуют. Байт синхронизации - это байт, содержащий заранее известный код, например 0111110, который оповещает приемник о приходе кадра данных. Его обычно называют флагом. При его получении приемник должен войти в байтовый синхронизм с передатчиком, то есть правильно понимать начало очередного байта кадра. Иногда применяется несколько синхробайт для обеспечения более надежной синхронизации приемника и передатчика. Так как при передаче длинного кадра у приемника могут появиться проблемы с синхронизацией бит, то в этом случае используются самосинхронизирующие коды.



# Скорость передачи данных

Боды / Байты

Промежуточные устройства

Преобразование / шлюзование

# Вывод информации

Текстовый и графический режимы

Растровые и Векторные шрифты

# Текстовый режим

**Текстовый видеорежим** — режим компьютерного видеоадаптера, в котором экран представлен в виде решётки знакомест (а не пикселей, в отличие от графических режимов). В каждом из знакомест может находиться один символ из ограниченного набора.

Текстовые видеотерминалы начали заменять телетайпы в начале 1970-х годов и изменили способ ведения диалога оператора с компьютером. Вместо командной строки появился текстовый интерфейс пользователя; в шрифты начали вводить псевдографические символы для рисования рамок и имитации элементов графического интерфейса.

**Текстовый пользовательский интерфейс, ТПИ** (англ. **Text user interface, TUI**; также **Character User Interface, CUI**) — разновидность интерфейса пользователя, использующая при вводе-выводе и представлении информации исключительно набор буквенно-цифровых символов и символов псевдографики.

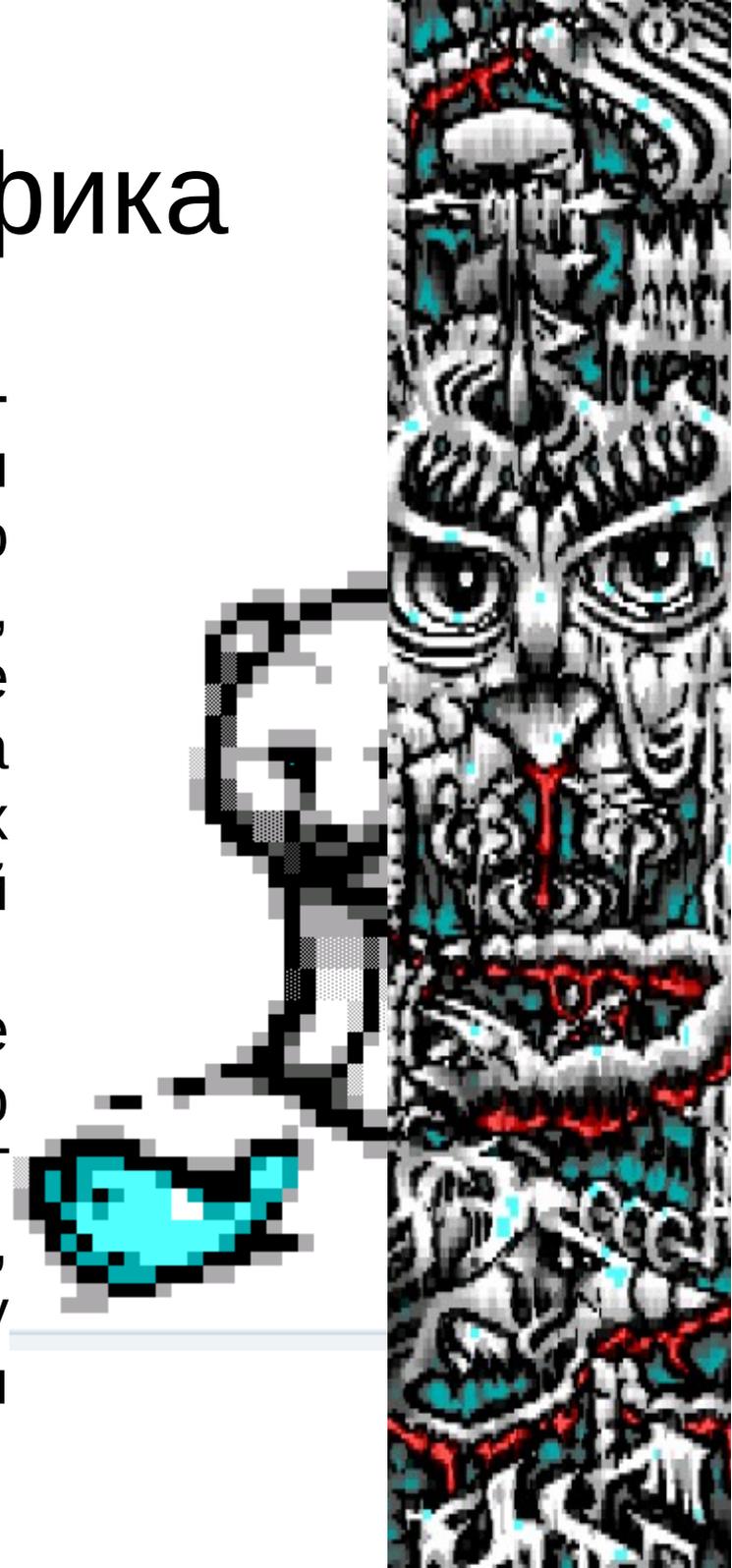
# Текстовый режим

Разре- шение	Размер символа	Графическое разрешение	Кол-во цветов	Адаптеры
80x25	9x14	720x350	Чёрно-белый	MDA, Hercules
40x25	8x8	320x200	16 цветов	CGA, EGA
80x25	8x8	640x200	16 цветов	CGA, EGA
80x25	8x14	640x350	16 цветов	EGA
80x43	8x8	640x350	16 цветов	EGA
80x25	9x16	720x400	16 цветов	VGA
80x30	8x16	640x480	16 цветов	VGA
80x50	9x8	720x400	16 цветов	VGA
80x60			16 цветов	VESA-совместимые Super VGA
132x25			16 цветов	VESA-совместимые Super VGA
132x43			16 цветов	VESA-совместимые Super VGA
132x50			16 цветов	VESA-совместимые Super VGA
132x60			16 цветов	VESA-совместимые Super VGA

# ASCII/ANSI графика

ANSI-графика — расширение ASCII-графики. Этот вид цифровой графики создаёт картинку из символов, но использует не только символы, предлагаемые кодировкой ASCII, а все 224 печатных символа, 16 цветов шрифта и 8 фоновых цветов, поддерживаемых драйвером ANSI.SYS, который использовался в системе DOS.

Среды DOS под Windows NT не используют ANSI.SYS; даже просмотр ANSI-графики в среде Windows NT требует специальных программ. Однако, графика ANSI по-прежнему поддерживается консольным текстовым драйвером Linux.



# Графический режим

Графический интерфейс пользователя (ГИП), графический пользовательский интерфейс (ГПИ) (англ. Graphical user interface, GUI) — разновидность пользовательского интерфейса, в котором элементы интерфейса (меню, кнопки, значки, списки и т. п.), представленные пользователю на дисплее, исполнены в виде графических изображений.

Благодаря исследованиям, проведённым в 60-е годы Дагом Энгельбартом в Стэнфорде был изобретён GUI. Впоследствии концепция GUI была перенята учеными из исследовательской лаборатории Xerox PARC в 1970-х. В 1973 году в лаборатории Xerox PARC появляется концепция графического интерфейса WIMP (Windows, Icons, Menus, Point-n-Click).

В 1973-1979 создаются экспериментальные компьютеры с GUI. Коммерческое воплощение концепция GUI получила в продуктах корпорации Apple Computer. В операционной системе AmigaOS GUI с многозадачностью был использован в 1985 году.

# Шрифты

Компьютерные шрифты делятся по способу отрисовки на два типа: растровые и векторные.

- В растровых шрифтах каждый символ описан в виде набора точек (пикселей), расположенных в узлах сетки растра — то есть по сути является обычным точечным рисунком. Растровые шрифты непригодны для высококачественной печати и используются в основном в программах с текстовым интерфейсом и в консоли.

Растровый шрифт

# Шрифты

- В векторных (или контурных) шрифтах символы представляют собой криволинейные контуры, описываемые математическими формулами. Каждый знак описан с помощью векторов, определяющих координаты опорных точек, которые соединены прямыми или кривыми и образуют контур знака без привязки к абсолютному размеру или разрешению. Такое описание позволяет легко изменять масштаб изображения без потери качества, что невозможно в случае с растровыми шрифтами. Векторные шрифты одинаково выглядят как на экране, так и на бумаге. Существует несколько различных форматов векторных шрифтов, различающихся способом хранения и представления информации о шрифте: PostScript Type1, TrueType, OpenType.

Для вывода векторного шрифта на растровые устройства (мониторы и принтеры) его необходимо растеризовать — преобразовать в набор точек.

# Шрифты

По ширине символов шрифты делят на моноширинные и пропорциональные.

- В моноширинном шрифте все символы имеют одинаковую ширину. В ранних компьютерах это позволяло упростить задачу вывода текста на экран: каждый символ располагался в пределах своего знакоместа, поэтому не требовалось задание ширины символа, и процедура задания положения символа на экране становилась чрезвычайно простой. Другим достоинством моноширинных шрифтов является однозначность относительного расположения символов на экране вне зависимости от гарнитуры шрифта, что является необходимым условием для печати с преформатированием. Моноширинные шрифты применяются прежде всего в эмуляторах терминалов, а также в интерфейсах сред разработки.

- *В пропорциональном шрифте символы могут иметь разную ширину. К примеру буква l будет занимать значительно меньше места, чем буква W.*