



ФГОБУ ВПО "СибГУТИ"
Кафедра вычислительных систем

ПРОГРАММИРОВАНИЕ ЯЗЫКИ ПРОГРАММИРОВАНИЯ

ВВОДНАЯ ЛЕКЦИЯ

Преподаватель:

Доцент Кафедры ВС, к.т.н.

Поляков Артем Юрьевич



Структура курса

Продолжительность	2 семестра (34 учебные недели)
Виды занятий	<ul style="list-style-type: none">▪ Лекционные (34 занятия);▪ Лабораторные работы (34 занятия).
Итоговый контроль	<ul style="list-style-type: none">▪ зачет (1 семестр)▪ экзамен (2 семестр)▪ курсовой проект (2 семестр)
Семестр 1	Изучение основ программирования. Форма практических занятий: решение учебных задач
Семестр 2	Изучение алгоритмов и динамических структур данных. Получение навыков решения прикладных задач. Форма практических занятий: выполнение лабораторных курсовой работ.



План занятий в осеннем семестре

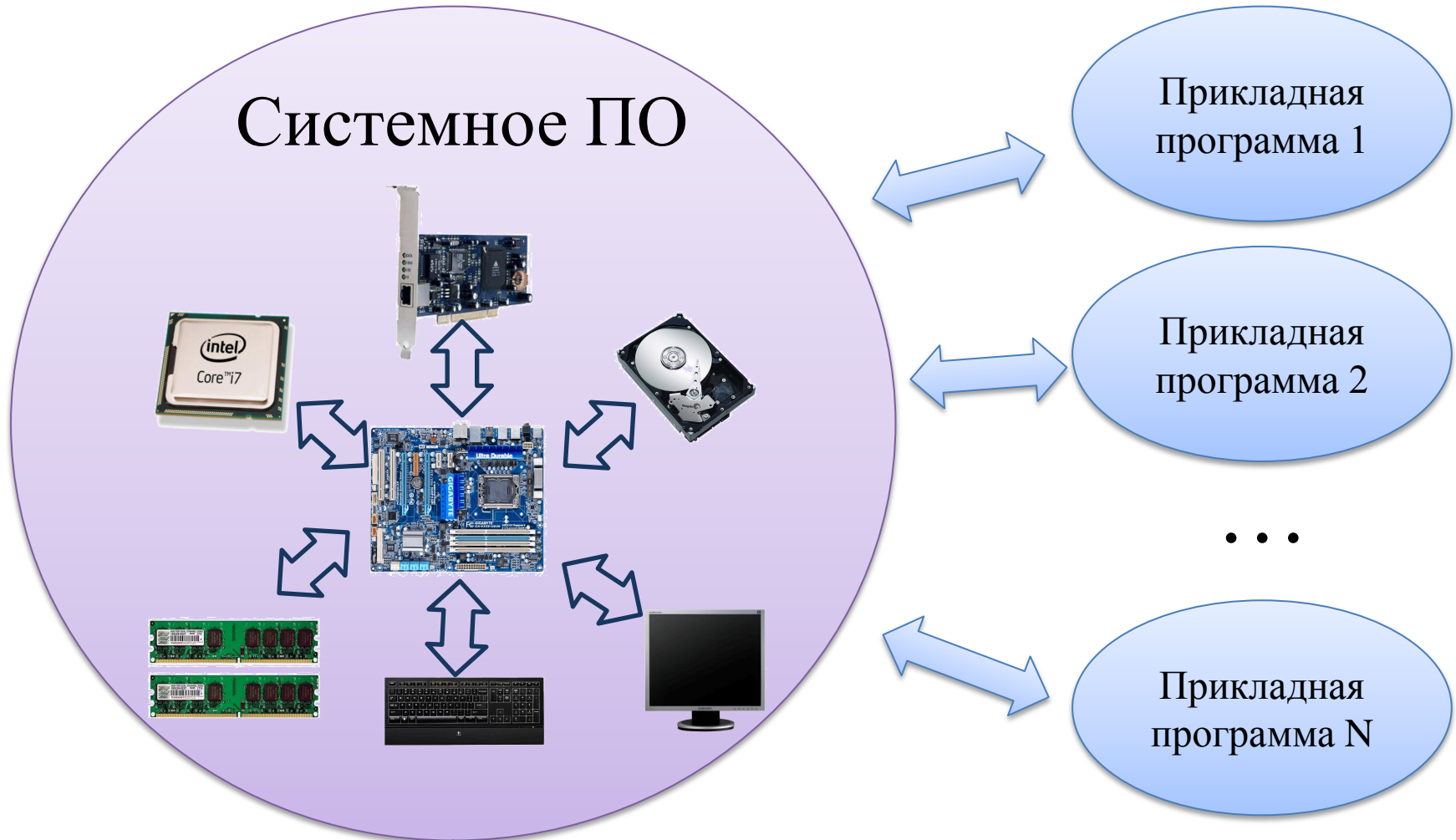
1. Введение в программирование.
2. Основные типы данных и управляющие конструкции языка СИ.
3. Обработка массивов данных.
4. Модульное программирование.
5. Обработка табличных данных.
6. Работа с текстовыми и бинарными файлами.
7. Преобразование типов данных, работа с указателями.
8. Классы памяти.

Аппаратное обеспечение ВЫЧИСЛИТЕЛЬНЫХ МАШИН





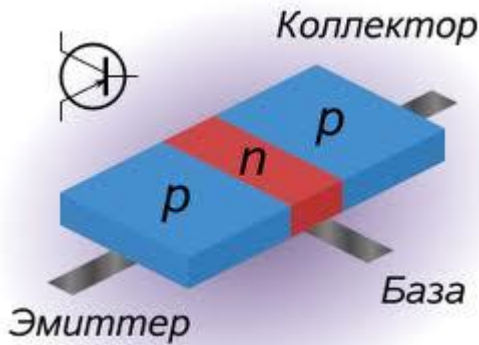
Программное обеспечение вычислительных машин



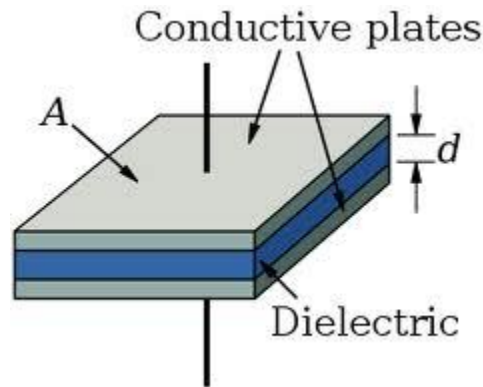


Хранение информации в вычислительных машинах

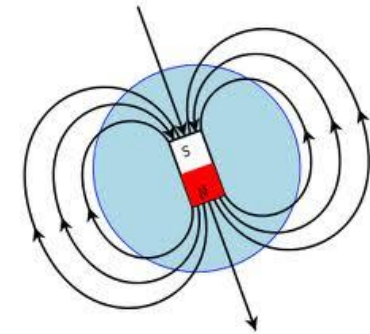
Основная задача вычислительных средств –
хранение и обработка информации



Транзистор



Конденсатор



Ферромагнетики

Элементная база позволяет создавать приборы,
имеющие два устойчивых состояния

Наиболее удобной является двоичная система счисления



Двоичная система счисления

Цифры двоичной СС: **0, 1**

x_{10}	x_2
0	0
1	1
2	10
3	11
4	100
5	101
6	110
7	111

$$4_{10} = 3_{10} + 1 = 11_2 + 1 = 100_2$$

x_{10}	x_2
8	1000
9	1001
10	1010
11	1011
12	1100
13	1101
14	1110
15	1111



Связь двоичной и шестнадцатеричной СС

Родственные СС, один шестнадцатеричный разряд представляется четырьмя двоичными

x_2	x_{10}	x_{16}
0000	0	0
0001	1	1
0010	2	2
0011	3	3
0100	4	4
0101	5	5
0110	6	6
0111	7	7

x_2	x_{10}	x_{16}
1000	8	8
1001	9	9
1010	10	A
1011	11	B
1100	12	C
1101	13	D
1110	14	E
1111	15	F



Связь двоичной и шестнадцатеричной СС (2)

x_2	x_{10}	x_{16}
0000	0	0
0001	1	1
0010	2	2
0011	3	3
0100	4	4
0101	5	5
0110	6	6
0111	7	7

x_2	x_{10}	x_{16}
10000	16	10
10001	17	11
10010	18	12
10011	19	13
10100	20	14
10101	21	15
10110	22	16
10111	23	17



Перевод $x_2 \rightarrow x_{16}$

x_2	x_{16}	x_2	x_{16}
0000	0	1000	8
0001	1	1001	9
0010	2	1010	A
0011	3	1011	B
0100	4	1100	C
0101	5	1101	D
0110	6	1110	E
0111	7	1111	F

Для перевода из двоичной СС в шестнадцатеричную достаточно разбить x_2 на 4-хразрядные блоки и перевести каждый из них по отдельности:

$$\begin{aligned} &11011010101_2 = \\ &\mathbf{0110\ 1101\ 0101}_2 = \\ &\mathbf{6D5}_{16} \end{aligned}$$



Перевод $x_{16} \rightarrow x_2$

x_2	x_{16}	x_2	x_{16}
0000	0	1000	8
0001	1	1001	9
0010	2	1010	A
0011	3	1011	B
0100	4	1100	C
0101	5	1101	D
0110	6	1110	E
0111	7	1111	F

Для перевода из шестнадцатеричной СС в двоичную необходимо каждый разряд шестнадцатеричного числа представить 4-хразрядным двоичным числом:

$$2F8_{16} = 0010\ 1111\ 1000_2$$



Единицы измерения информации

БИТ – *binary digit* – разряд двоичного числа.

1

или

0

БАЙТ – минимально адресуемый набор из **8 битов**.

1	0	1	0	1	0	1	0
---	---	---	---	---	---	---	---

Килобайт (КБ) = 1024 байт

Мегабайт (МБ) = 1024 КБ

Гигабайт (ГБ) = 1024 МБ

Терабайт (ТБ) = 1024 ГБ

Петабайт (ПБ) = 1024 ТБ



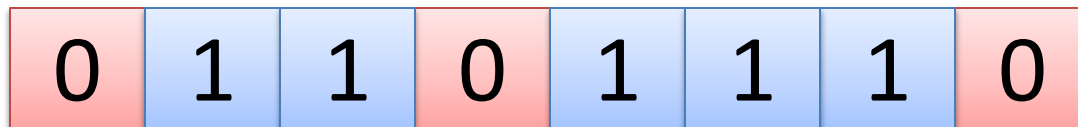
Внутреннее представление беззнаковых целых чисел

Хранение чисел в ЭВМ базируется
на их **двоичном представлении**.

Различают **знаковые** и **беззнаковые** целые, а также
вещественные числа.

Беззнаковые (unsigned) целые числа в памяти
хранятся в виде двоичного представления исходного
целого числа, например:

$$110_{10} = 1101110_2 = 01101110_2$$





Представление текстовой информации

Хранение информации возможно
ТОЛЬКО В ВИДЕ ДВОИЧНЫХ ЧИСЕЛ.

Текст **не является** числом или набором чисел!

Требуется **способ** преобразовать текст
в набор чисел (оцифровать)

Текст имеет естественное разбиение на:

- 1. слова**
- 2. буквы**



Представление текстовой информации (2)

- Текст кодируется посимвольно.
- Каждому символу сопоставляется уникальное число. Данное отображение называется кодировкой. Пример: таблица ASCII-кодов.

<i>Симв.</i>	<i>Код₁₆</i>	<i>Код₁₀</i>
'a'	61	98
'b'	62	99
'c'	63	100
'd'	64	101
'e'	65	102
'f'	66	103

<i>Симв.</i>	<i>Код₁₆</i>	<i>Код₁₀</i>
'1'	30	48
'2'	31	49
'3'	32	50
'4'	33	51
'5'	34	52
'6'	35	53



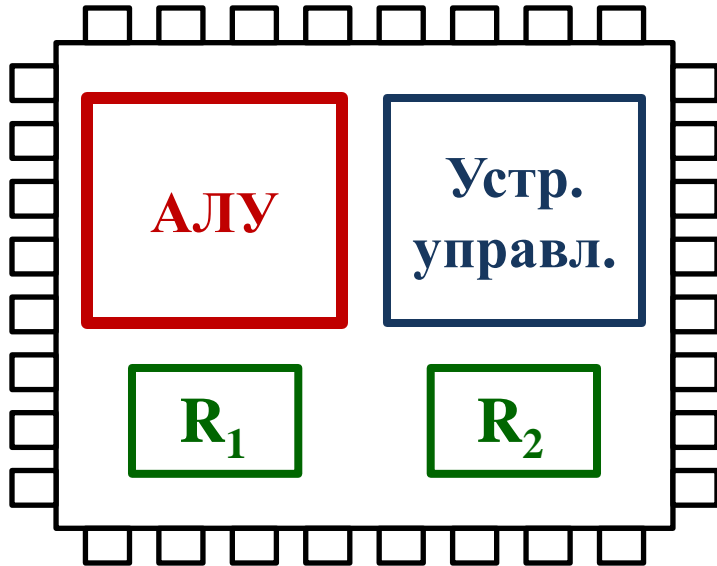
Представление текстовой информации (3)

- Полная версия таблицы ASCII-кодов.
- Коды указаны в шестнадцатеричной системе счисления

	.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	TAB	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



Обработка информации (упрощенный процессор)



<i>Команда</i>	<i>Код</i>	<i>Описание</i>
ADD	01	$R_1 + R_2$
SUB	02	$R_1 - R_2$
MUL	03	$R_1 * R_2$
DIV	04	R_1 / R_2

<i>Регистр</i>	<i>Код</i>	<i>Описание</i>
R_1	01	Регистры общего назначения
R_2	02	



Специфика обработки информации на современных процессорах



1. Количество регистров не позволяет хранить большие объемы информации
2. Арифметико-логическое устройство (АЛУ) служит для выполнения доступных команд.
3. Устройство управления обеспечивает выполнение команд

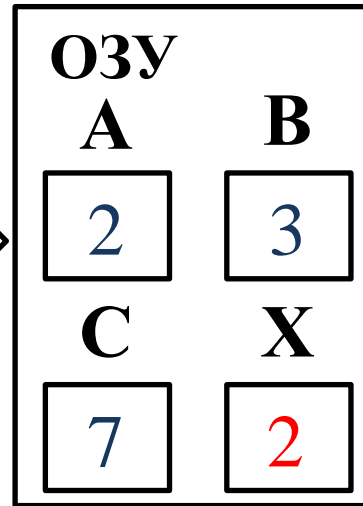


Задача поиска корней уравнения вида: $a \cdot x + b = c$

1. Решение задачи математически. Выражение неизвестной переменной x через известные переменные a , b и c :
 $x = (c - b)/a$
2. Реализация вычислений по полученной формуле на упрощенном процессоре
3. Входные данные (значения a , b и c) вводятся с клавиатуры или из файла и размещаются в оперативной памяти в ячейках **A** (адрес **10**), **B(20)** и **C(30)**. Адрес – порядковый номер первого байта ячейки.
4. Результат записывается в ячейку **X** с адресом 40.



Вычисление $x = (c - b)/a$ на упрощенном процессоре



SUB	01	R1	01
DIV	02	R2	02
MOV	03		
A	10		
B	20		
C	30		
X	40		

Машинное слово	Комментарий
----------------	-------------

03 30 01	MOV C -> R1
----------	-------------

03 20 02	MOV B -> R2
----------	-------------

01	SUB (R1 = R1 - R2)
----	--------------------

03 10 02	MOV A -> R2
----------	-------------

02	DIV (R1 = R1 / R2)
----	--------------------

Машинное слово	Комментарий
----------------	-------------

03 02 40	MOV R2 -> X
----------	-------------



Программа на низкоуровневом языке ASSEMBLER

Машинное слово	Программа на языке ASSEMBLER
03 30 01	MOV C R1
03 20 02	MOV B R2
01	SUB
03 10 02	MOV A R2
02	DIV
03 02 40	MOV R2 X

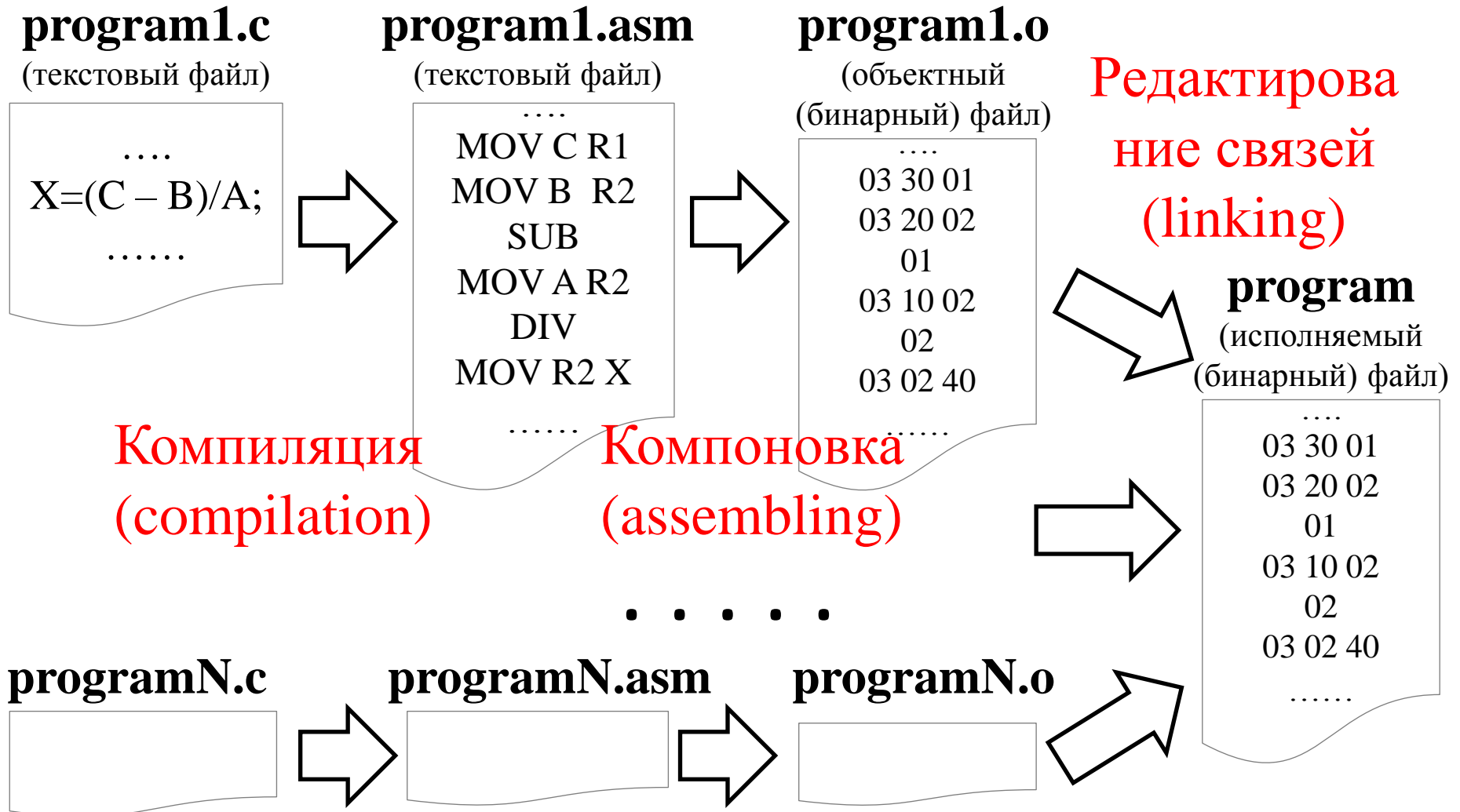


Программа на языке высокого уровня Си

Машинное слово	Программа на языке ASSEMBLER	Программа на языке Си
03 30 01	MOV C R1	$X = (C - B) / A;$
03 20 02	MOV B R2	
01	SUB	
03 10 02	MOV A R2	
02	DIV	
03 02 40	MOV R2 X	



Этапы формирования исполняемого кода из исходного





СПАСИБО ЗА ВНИМАНИЕ!