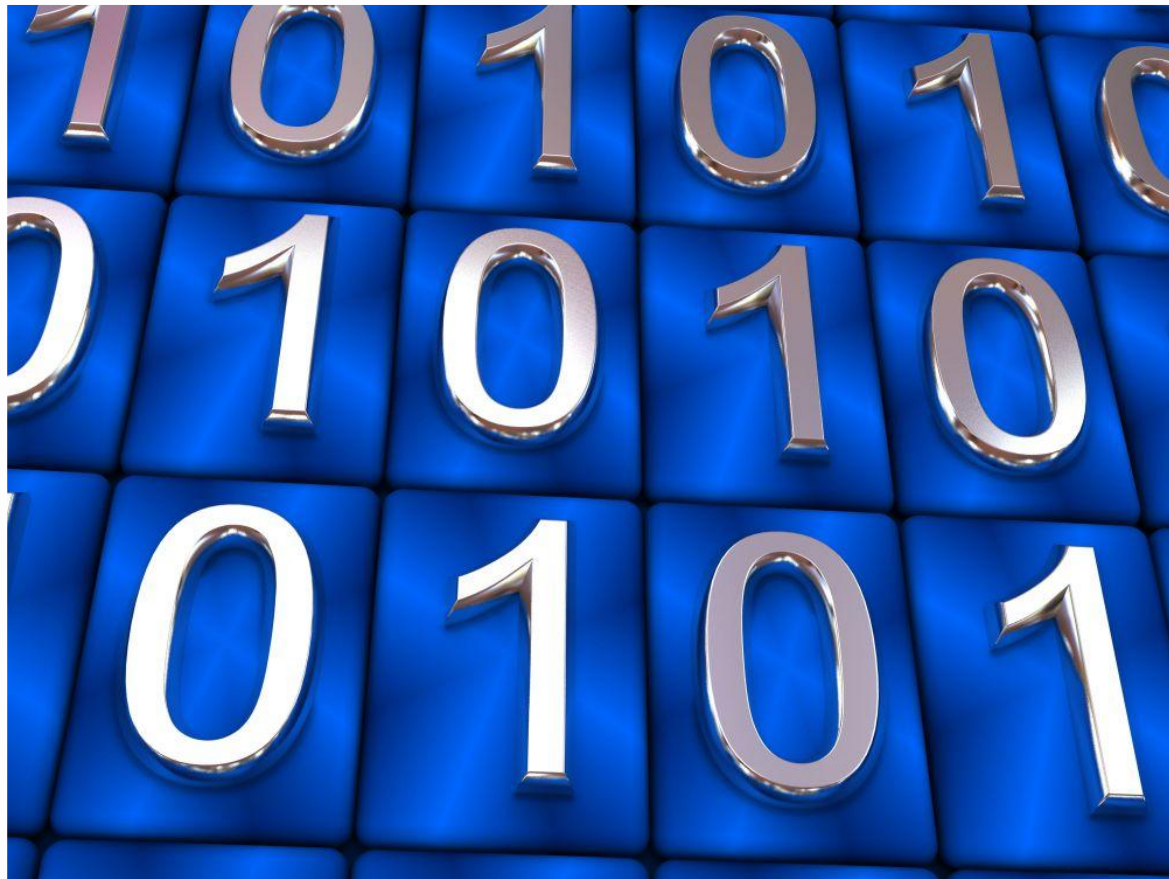
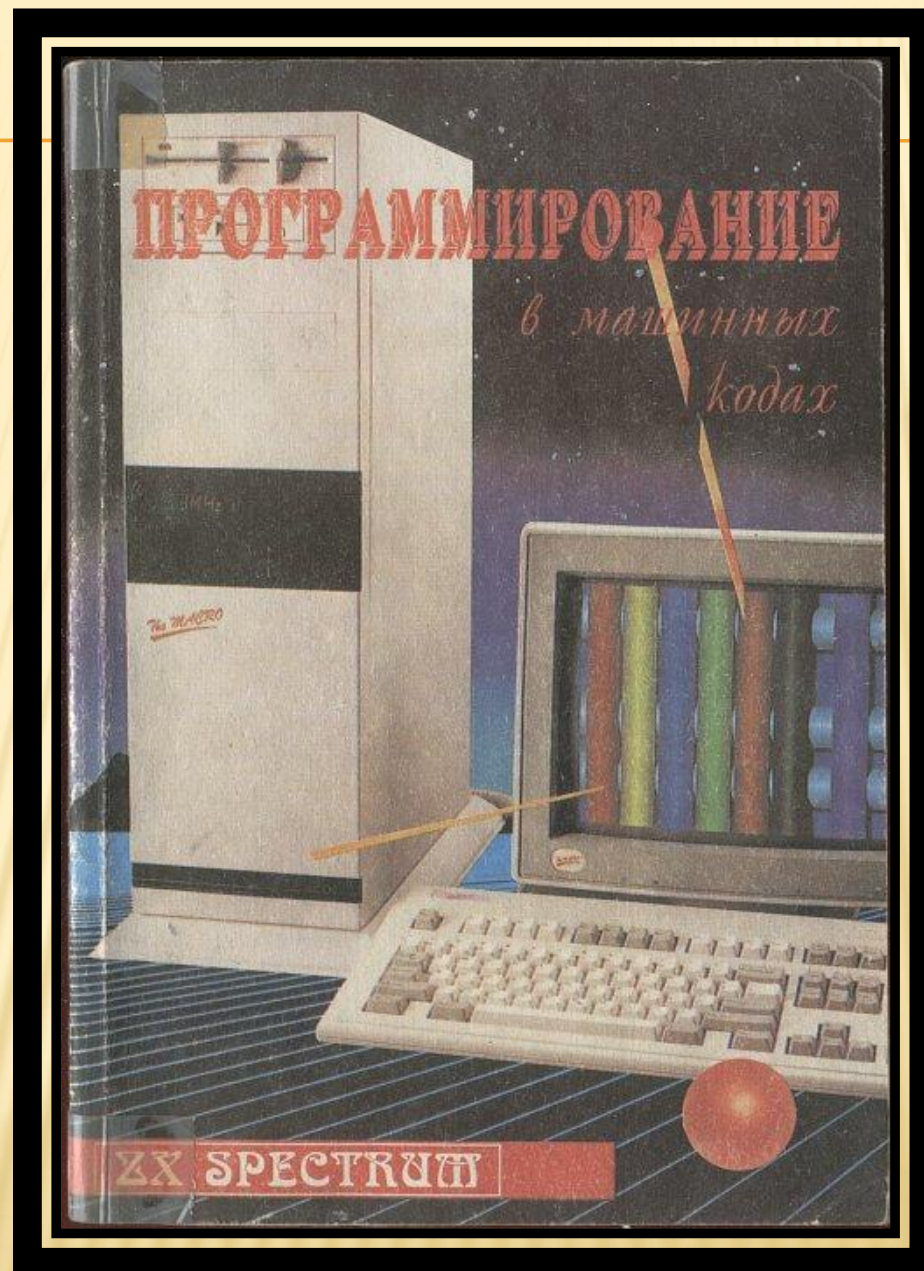


- Язык программирования — формальная знаковая система, предназначенная для записи компьютерных программ. Язык программирования определяет набор лексических элементов и правил их комбинации.



ПЕРВЫЕ УНИВЕРСАЛЬНЫЕ ЯЗЫКИ

Первые программы писались на машинном языке. Программисты обязаны были знать архитектуру машины досконально. Программы были достаточно простыми, что обуславливалось, во-первых, весьма ограниченными возможностями этих машин, и, во-вторых, большой сложностью разработки и, главное, отладки программ непосредственно на машинном языке. Вместе с тем такой способ разработки давал программисту просто невероятную власть над системой. Становилось возможным использование хитроумных алгоритмов и способов организации программ. Например, могла применяться такая возможность, как самомодифицирующийся код. Знание двоичного представления команд позволяло иногда не хранить некоторые данные отдельно, а встраивать их в код как команды.



ПРОГРАММИРОВАНИЕ

*в машинных
кодах*

The MACRO

ZX СПЕКТРУМ

СИСТЕМА КОМАНД ЭВМ-220

Операции над числами (Р, Q, T - порядки чисел X, Y, Z по A ₁ , A ₂ , A ₃)					Логические операции (F, D, S - значения кодов по A ₁ , A ₂ , A ₃)					Операции сдвига					Команды передачи управления с изменением [РА]									
Операция	КОП	Результат	ω-1	t мксек	Операция	КОП	Результат	ω-1	t мксек	Операция	КОП	Результат сдвига	ω-1	t мксек	КОП	Код в РА	Передача управл в A ₂	Передача управл в КРА+1	t мксек					
Сложение	01	21 41 61			Сравнение	15	$C_n = \bar{F}_n \cdot \bar{Q}_n$			Сдвиг кода по A ₁	54	$C_n = \bar{P}_n; S = \bar{P}_n \cdot 100$	C=0	24/155	11		РА < A ₁ и ω=1	РА > A ₁ или ω=0						
Вычитание	02	22 42 62	Z < 0	285	Сравнение с ост	35	аналогично 15, но по DCT при C=0		24	Сдвиг кода по P	74	$C_n = \bar{P}_n; S = P \cdot 100$	C=0	24/155	31		РА > A ₁ и ω=1	РА < A ₁ или ω=0						
Вычитание модулей	03	23 43 63			Логическое умножение	55	$C_n = \bar{F}_n \cdot \bar{Q}_n$			Сдвиг мантиссы по P	14	$C_n = \bar{P}_n; S = A_1^{10} \cdot 100$	C=0	24/155	51		РА < A ₁ и ω=0	РА > A ₁ или ω=1						
Деление	04	24 - -			Логическое сложение	75	$C_n = \bar{F}_n \vee \bar{Q}_n$			Сдвиг мантиссы по P	34	$C_n = \bar{P}_n; S = P \cdot 100$	C=0	24/155	71		РА > A ₁ и ω=0	РА < A ₁ или ω=1						
Умножение	05	25 45 65	Y < 100	51	Специальные операции над кодами					Операции засылки					Команды передачи управления с изменением в A ₃									
Извлечение кв. корня	44	64 - -		272	Цикл сложения	07	$C^i = (F^i + 1) \text{mod } 2^m$			Операция	КОП	Результат	ω-1	t мксек	12		РА < A ₁	РА > A ₁	24					
Выход младших разрядов произведения	47			24	Цикл вычитания	27	$C^i = (F^i - 1) \text{mod } 2^m$			Выборка из динамики	17	Засылка кода [Z] → Y у-ячейка по A ₂			40		РА > A ₁	РА < A ₁						
Операции изменения порядка					Сложение мантисс	13	$C^i = F^i$			Засылка в динамику	37	Засылка кода [Y] → Z у-ячейка по A ₂			60		РА < зона A ₂ [A ₁]	РА > зона A ₂ [A ₁]						
Операция	КОП	Результат	ω-1	t мксек	Вычитание мантисс	33	$C^i = \bar{F}^i$		24	Засылка кода	00	[A ₁] → A ₂ A ₂ безразличен			Команды передачи управления с занесением в A ₃									
Сложение порядка с A ₁	06	$r = q + (A_1^{10} - 100)$			Сложение КОПов	53	$C^i = (F^i + 1) \text{mod } 2^m$			Засылка кода с КЗУ	20	[КЗУ] → A ₃ Эта запись по A ₁ ; N ₃ за A ₂		24	КОП	Код заносимый в A ₃	Передача управления в A ₂	Передача управления в КРА+1	t мксек					
Сложение порядка с P	26	$r = q + (P - 100)$	r < 100	24	Вычитание КОПов	73	$C^i = (F^i - 1) \text{mod } 2^m$			Анализ окончания раб АМ	20	A ₁ = 0000 - переход на N ₃ при 0 Траб АМ			16	0160000 A ₁ 0000	всегда	-						
Вычитание порядка из A ₁	46	$r = q - (A_1^{10} - 100)$			Цикл сдвиг	67	$C^i = F^i \text{mod } 2^m$			Команды передачи управления с изменением [РА]														
Вычитание порядка из P	66	$r = q - (P - 100)$			Команда останова					Команды ввода														
КОП Совмещенные операции					Операции в M2, выполняемые по сигналам из M1					Операции, выполняемые по командам Ma и Mb					Ma (КОП 50)					Mb (КОП 70)				
77	Вызов [A ₁] и [A ₂] на P ₁ и P ₂ гашение [A ₃] при проболжени				Операция					Режим					A ₂					A ₃				
КОП	Совмещенные операции				Выборка команд в M2 по адресу из M1					УЧ					12 11 10 9 8 7 6 5 4 3 2 1					A ₂ A ₃ t мксек				
77	Вызов [A ₁] и [A ₂] на P ₁ и P ₂ гашение [A ₃] при проболжени				Информация переданная на РБФ M2 из M1					УЧ					12 11 10 9 8 7 6 5 4 3 2 1					A ₂ A ₃ t мксек				
52	A ₂	0520000 A ₁ 0000			40÷38р 34р 24÷13р					0010					12 11 10 9 8 7 6 5 4 3 2 1					A ₂ A ₃ t мксек				
72	зона A ₂ [A ₂]	0520000 A ₁ 0000			P1 адреса без адрес в M2 контр команды					0020					12 11 10 9 8 7 6 5 4 3 2 1					A ₂ A ₃ t мксек				
Команды ввода					Обмен кодами M1 ↔ M2					0500					12 11 10 9 8 7 6 5 4 3 2 1					A ₂ A ₃ t мксек				
Операция	КОП	A ₁	A ₂	A ₃	40÷38р 36÷25р 24÷13р					0100					12 11 10 9 8 7 6 5 4 3 2 1					A ₂ A ₃ t мксек				
Ввод с остановом при несоблюдении КΣ	10	A ₁ M3У	KPA	AKΣ	P1A2M2 УЧ АН M3У					0140					12 11 10 9 8 7 6 5 4 3 2 1					A ₂ A ₃ t мксек				
Ввод без ост. при несоблюд КΣ	30	A ₁ M3У	KPA	AKΣ	Операции выполняемые по сигналам от линии связи и аналоговой машины					0200					12 11 10 9 8 7 6 5 4 3 2 1					A ₂ A ₃ t мксек				
Операция изменения регистров приращеня					Операция					Накопление					12 11 10 9 8 7 6 5 4 3 2 1					A ₂ A ₃ t мксек				
КОП	A ₁	A ₂	код заносимый в A ₃		Авторазрыв от ЛС					0300					12 11 10 9 8 7 6 5 4 3 2 1					A ₂ A ₃ t мксек				
57	2,1р-РПМ1				Выборка команды из яч-ки 0022, 0 M3У без занесения 0022 на КРА					0040					12 11 10 9 8 7 6 5 4 3 2 1					A ₂ A ₃ t мксек				
	54р-РПМЕ	32,1р-РПА3			Выборка команды из яч-ки 0023, 0 M3У без занесения 0023 на КРА					0060					12 11 10 9 8 7 6 5 4 3 2 1					A ₂ A ₃ t мксек				
	7р-Пр А3	634р-РПА2			Засылка состояния					0030					12 11 10 9 8 7 6 5 4 3 2 1					A ₂ A ₃ t мксек				
	8р-Пр А2	987р-РПА1			Засылка состояния					0034					12 11 10 9 8 7 6 5 4 3 2 1					A ₂ A ₃ t мксек				
	9р-Пр А1	2,11,10р-РПАМ			Выборка команды в M2 по адресу из M1					0033					12 11 10 9 8 7 6 5 4 3 2 1					A ₂ A ₃ t мксек				
	10р-Пр КРА				Обмен кодами M1 ↔ M2					0031					12 11 10 9 8 7 6 5 4 3 2 1					A ₂ A ₃ t мксек				
	11р-Пр МП				Работа с АМ					0032					12 11 10 9 8 7 6 5 4 3 2 1					A ₂ A ₃ t мксек				
	12р-Пр МБ														12 11 10 9 8 7 6 5 4 3 2 1					A ₂ A ₃ t мксек				

1. Емкость промежуточного накопителя вывода ПНВ 1024 слова.

2. После вывода содержимое ПНВ не стирается.

3. При обмене с МБ возможен переход с данного блока на следующий (4096-4097).

4. Команда останова имеет код только 77.

5. Команда 20 при A₁=0000 используется как команда перехода по признаку работы с аналоговой машиной.

6. В НМЛ M3=0 используется только в режиме "Разметка" для залипания дефектных зон МП.

1. При чтении адреса с ПК 1^й р РБФ-блокировка АВ ост при несоблюдении КΣ

1^й 13р РБФ-блокировка записи в М3У последующего массива 37-38р РБФ(РПА) → → 13-14р СМ А

2. При A₁=0 блокируется запись в М3У последующего массива.

* Обращение к М3У по команде ИРП производится по старым значениям только по командам ИРП, АРЗВ

* Операции АР1M2, АР2M2 АРАС, АРАМ не изменяют состояния машины.

Примечание: Знак, X означает, что состояние разряда безразлично. Отсутствие знак, означает, что в разряде может быть 0 или 1; A₁-адрес исполнительный; [A₁]-содержимое ячейки с адресом A



Первым значительным шагом представляется переход к языку ассемблера. Программисту не надо было больше вникать в способы кодирования команд на аппаратном уровне. Появилась также возможность использования макросов и меток, что также упрощало создание, модификацию и отладку программ.

АССЕМБЛЕР

Вместе с тем, переход к новому языку таил в себе и некоторые отрицательные стороны. Возможности программистов сильно сократились. Кроме того, здесь впервые в истории развития программирования появились два представления программы: в исходных текстах и в откомпилированном виде. К концу ассемблерной эры возможность автоматической трансляции в обе стороны была утеряна. В связи с этим было разработано большое количество специальных программ-дизассемблеров, осуществляющих обратное преобразования, однако в большинстве случаев они с трудом могут разделить код и данные.

Data Analyzer 1.X

Model GeoExtent Options Run State Browser Help



Input Definition

Set Data Set

Model: max/min range generator
Variable: cover (1 layers)
Units: CLASS
Source: Not Set

OK

Model Definition

Set Input Variable

Filter
1/localdb/biophys/lc/global/30min/olson/dm/*

Directories

- on/dm/.
- on/dm/..

Files

- M:clolsonveg2-72.cat-1.0

Selection

1/localdb/biophys/lc/global/30min/olson/dm/

OK Filter Cancel Help

ФОРТРАН

Следующий шаг был сделан в 1954 году, когда был создан первый язык высокого уровня — Фортран. Впервые программист мог по-настоящему абстрагироваться от особенностей машинной архитектуры. Синтаксическая структура языка была достаточно сложна для машинной обработки в первую очередь из-за того, что пробелы как синтаксические единицы вообще не использовались. Это порождало массу возможностей для скрытых ошибок, таких, например:

В Фортране конструкция : “DO 10 I=1,100” описывает «цикл выполнения оператора при изменении индекса от 1 до 100» Если же здесь заменить запятую на точку, то получится оператор присваивания: DO10I = 1.100.

ФОРТРАН

Язык Фортран использовался для научных вычислений. Он страдает от отсутствия многих привычных языковых конструкций и атрибутов, компилятор практически никак не проверяет синтаксически правильную программу с точки зрения корректности. По признанию самого Бэкуса, перед ними стояла задача скорее разработки компилятора, чем языка. Понимание самостоятельного значения языков программирования пришло позже.



ФОРТРАН

Появление Фортрана было встречено еще более яростной критикой, чем внедрение ассемблера. Через некоторое время пришло понимание того, что реализация больших проектов невозможна без применения языков высокого уровня. Мощность вычислительных машин росла, и с тем падением эффективности, которое раньше считалось угрожающим, стало возможным смириться. Преимущества же языков высокого уровня стали настолько очевидными, что побудили разработчиков к созданию новых языков, все более и более совершенных.

-
- 1960 г. – создание языка Cobol
 - 1960 г. Петер Наур создал язык программирования Algol.
 - 1963 г. – создание языка BASIC
 - 1964 г. – корпорация IBM создала язык PL/1
 - 1968 г. – новая версия языка Algol.

PASCAL-ПОДОБНЫЕ ЯЗЫКИ



В 1970 году Никлаусом Виртом был создан язык программирования Pascal. Язык замечателен тем, что это первый широко распространенный язык для структурного программирования. В этом языке также внедрена строгая проверка типов, что позволило выявлять многие ошибки на этапе компиляции.

СИ-ПОДОБНЫЕ ЯЗЫКИ

- В 1972 году Керниганом и Ритчи был создан язык программирования Си. Через 14 лет Бьярн Страуструп создал первую версию языка C++, добавив в язык C объектно-ориентированные черты. Язык стал основой для разработки современных больших и сложных проектов. В 1999–2000 годах в корпорации Microsoft был создан язык C#. Он в достаточной степени схож с Java (и задумывался как альтернатива последнему), но имеет и отличительные особенности. Ориентирован, в основном, на разработку многокомпонентных Интернет-приложений.

- В 1969 году был создан язык SETL — язык для описания операций над множествами. Основной структурой данных в языке является множество, а операции аналогичны математическим операциям над множествами.
- Perl — язык создавался в помощь системному администратору операционной системы Unix для обработки различного рода текстов и выделения нужной информации. Развился до мощного средства работы с текстами.
- Python — интерпретируемый, объектно-ориентированный язык программирования. По структуре и области применения близок к Perl, однако менее распространен и более строг и логичен.

Спасибо за внимание!)