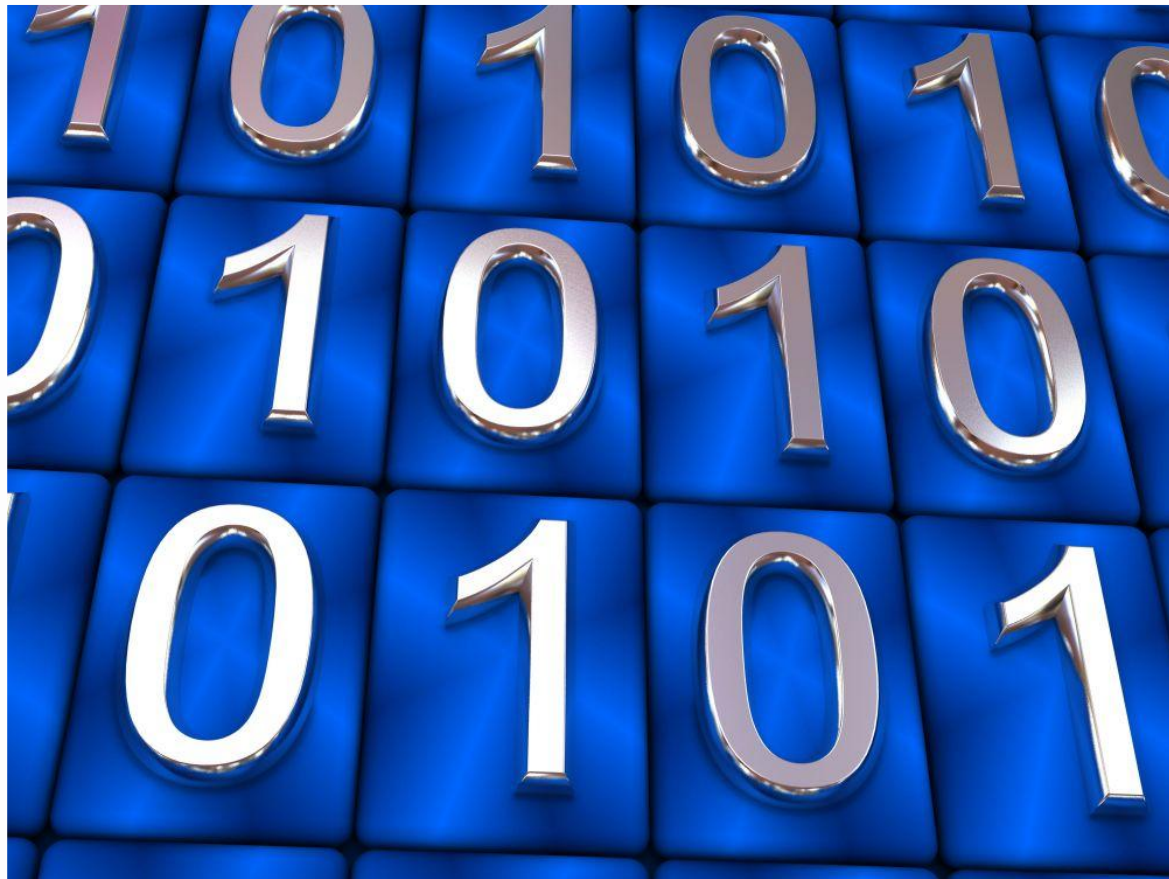
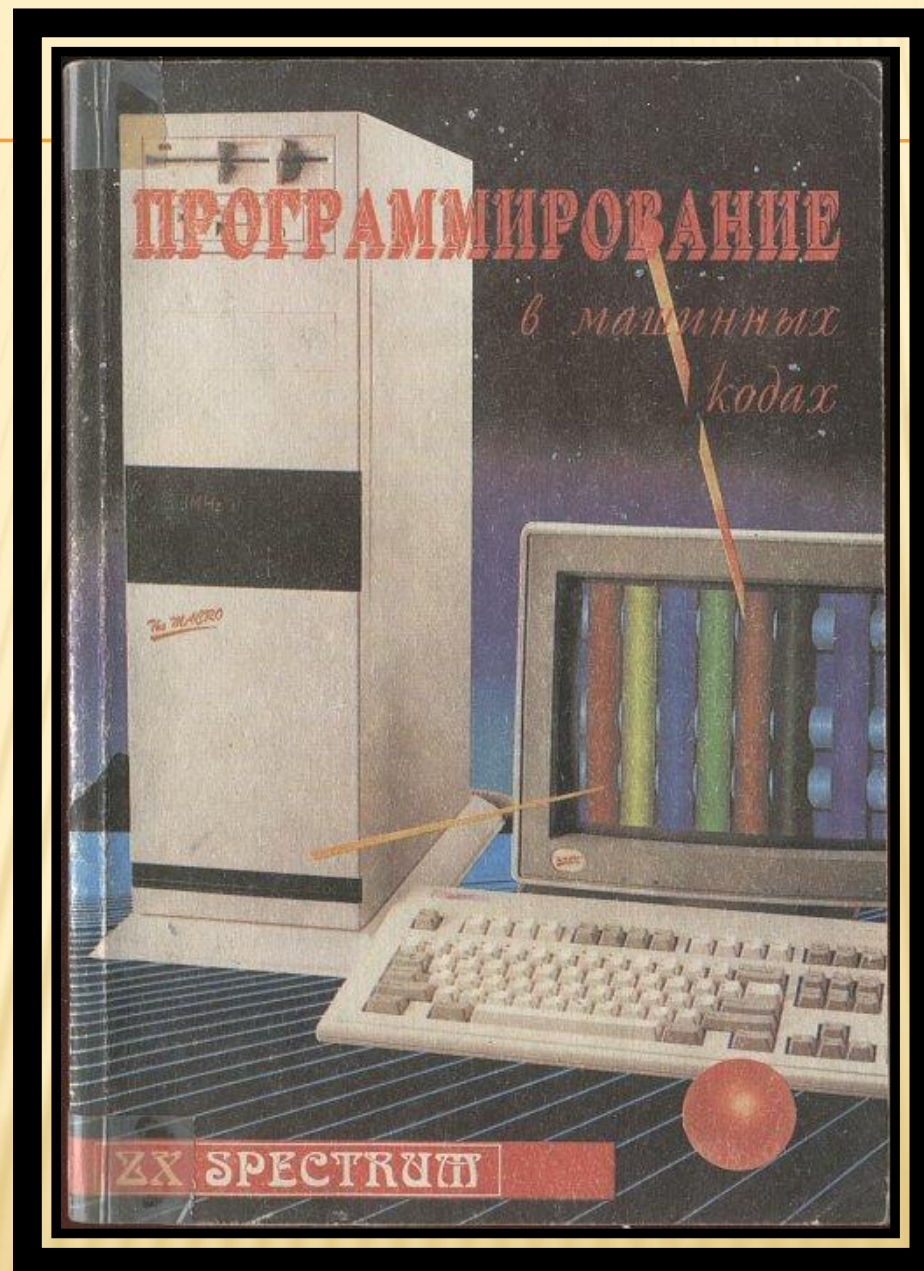


- Язык программирования — формальная знаковая система, предназначенная для записи компьютерных программ. Язык программирования определяет набор лексических элементов и правил их комбинации.



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.



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	$C_k = \bar{F}_k \cdot \bar{D}_k$			Сдвиг кода по A ₁	54	$C_{k+1} = \bar{D}_k; S = \bar{D}_k \cdot 100$			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 _k ≠ 0		24	Сдвиг кода по P	74	$C_{k+1} = \bar{D}_k; S = P \cdot 100$			31		РА > A ₁ и ω=1	РА < A ₁ или ω=0	
Вычитание модулей	03	23 43 63			Логическое умножение	55	$C_k = \bar{F}_k \cdot \bar{D}_k$			Сдвиг мантиссы по P	14	$C_{k+1} = \bar{D}_k; S = A_1^{10} \cdot 100$			51		РА < A ₁ и ω=0	РА > A ₁ или ω=1	
Деление	04	24 - -		-107	Логическое сложение	75	$C_k = \bar{F}_k \vee \bar{D}_k$			Сдвиг мантиссы по P	34	$C_{k+1} = \bar{D}_k; S = P \cdot 100$			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	$C^i = (F^i + 1) \bmod 2^m$			Операция	КОП	Результат	ω-1	t мксек	12		РА < A ₁	РА > A ₁	24
Выход младших разрядов произведения	47			24	Цикл вычитания	27	$C^i = (F^i - 1) \bmod 2^m$			Выборка из динамики	17	Засылка кода [Z] → Y у-ячейка по A ₂			32		РА > A ₁	РА < A ₁	
Операции изменения порядка					Сложение мантисс					Засылка в динамику					Команды передачи управления с занесением в A ₃				
Операция	КОП	Результат	ω-1	t мксек	Сложение мантисс	13	$C^i = F^i$			Засылка кода [Y] → Z у-ячейка по A ₂	37			40		РА < зона A ₂ [A ₁]	РА > зона A ₂ [A ₁]		
Сложение порядка с A ₁	06	$r = q + (A_1^{10} - 100)$			Вычитание мантисс	33	$C^i = F^i$		24	Засылка кода [A ₁] → A ₂ A ₂ безразличен	00			60		РА > зона A ₂ [A ₁]	РА < зона A ₂ [A ₁]		
Сложение порядка с P	26	$r = q + (P - 100)$	r > 100	24	Сложение КОПов	53	$C^i = (F^i + 1) \bmod 2^m$			Засылка кода с КЗУ	20	[КЗУ] → A ₃ Эта запись A ₃ ; N ₃ заду		24	16	010000 A ₃ 0000	всегда	-	
Вычитание порядка из A ₁	46	$r = q - (A_1^{10} - 100)$			Вычитание КОПов	73	$C^i = (F^i - 1) \bmod 2^m$			Анализ окончания раб АМ	20	A ₃ = 0000 - переход на N ₃ при 0 Траб АМ			56	[A ₁]	ω=1	ω=0	24
Вычитание порядка из P	66	$r = q - (P - 100)$			Цикл сдвиг	67	$C^i = F^i \bmod 2^m$ A ₂ безразличен			Ма (КОП 50)					МБ (КОП 70)				
Команда останова					Операции в M2, выполняемые по сигналам из M1					Операции, выполняемые по командам Ма и МБ									
КОП	Совмещенные операции				Операция					Режим					A ₂				
77	Вызов [A ₁] и [A ₂] на P ₁ и P ₂ гашение [A ₃] при проболжени				Выборка команд в M2 по адресу из M1					УЧ					A ₃				
52	A ₂	0520000 A ₁ 0000			Информация переданная на РБФ M2 из M1					12 11 10 9 8 7 6 5 4 3 2 1					A ₁ A ₂ A ₃				
72	зона A ₂ [A ₂]	0520000 A ₁ 0000			40÷38p 34p 24÷13p P1адреса без адрес в M2 контр команды					10 9 8 7 6 5 4 3 2 1 A ₂ A ₃ MЗУ					A ₁ A ₂ A ₃				
Команды ввода					Обмен кодами M1 ↔ M2					Обмен с МБ					A ₁ A ₂ A ₃				
Операция	КОП	A ₁	A ₂	A ₃	AP1 _{M2} 40÷38p 36÷25p 24÷13p P1A2M2 УЧ АНМЗУ					0010					A ₁ A ₂ A ₃				
Ввод с остановом при несоблюдении КΣ	10	A ₁ MЗУ	KPA	AKΣ	AP1 _{M2} P1A2M2 УЧ АНМЗУ					0020					A ₁ A ₂ A ₃				
Ввод без ост. при несоблюд КΣ	30	A ₁ MЗУ	KPA	AKΣ	Операции выполняемые по сигналам от линии связи и аналоговой машины					0500					A ₁ A ₂ A ₃				
Операция изменения регистров приращения					Операция					Печать „8“					A ₁ A ₂ A ₃				
КОП	A ₁	A ₂	код заносимый в A ₃		Авторазрыв от ЛС					0100					A ₁ A ₂ A ₃				
57	2,1p-РПМ1				AP1 _{M2} Выборка команды из яч-ки 0022, 0° MЗУ без занесения 0022 на КРА					0200					A ₁ A ₂ A ₃				
	54p-РПМЕ	32,1p-РПА3			AP1 _{M2} Выборка команды из яч-ки 0023, 0° MЗУ без занесения 0023 на КРА					0500					A ₁ A ₂ A ₃				
	7p-Пр А3	634p-РПА2			AP1 _{M2} Выборка команды из яч-ки 0023, 0° MЗУ без занесения 0023 на КРА					0100					A ₁ A ₂ A ₃				
	8p-Пр А2	987p-РПА1			AP1 _{M2} Выборка команды из яч-ки 0023, 0° MЗУ без занесения 0023 на КРА					0140					A ₁ A ₂ A ₃				
	9p-Пр А1	2,1,10p-РПАМ			AP1 _{M2} Выборка команды из яч-ки 0023, 0° MЗУ без занесения 0023 на КРА					0200					A ₁ A ₂ A ₃				
	10p-Пр КРА				AP1 _{M2} Выборка команды из яч-ки 0023, 0° MЗУ без занесения 0023 на КРА					0300					A ₁ A ₂ A ₃				
	11p-Пр МП				AP1 _{M2} Выборка команды из яч-ки 0023, 0° MЗУ без занесения 0023 на КРА					0040					A ₁ A ₂ A ₃				
	12p-Пр МБ				AP1 _{M2} Выборка команды из яч-ки 0023, 0° MЗУ без занесения 0023 на КРА					0060					A ₁ A ₂ A ₃				

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 MЗ=0 используется только в режиме „Разметка“ для залипания дефектных зон МЛ.

1. При чтении адреса с ПК 1° PБФ-блокировка АВ ост при несоблюдении КΣ

1° 13p PБФ-блокировка записи в МЗУ последующего массива 37-38p PБФ(РПА) → → 13-14p СМ А

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

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

* Операции AP1_{M2}, AP2_{M2}, AP1_{M2}, AP1_{M2} 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; [A] - 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
1localdb/biophys/lc/global/30min/olson/dm/*

Directories	Files
on/dm/.	M:clolsonveg2-72.cat-1.0
on/dm/..	

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!)