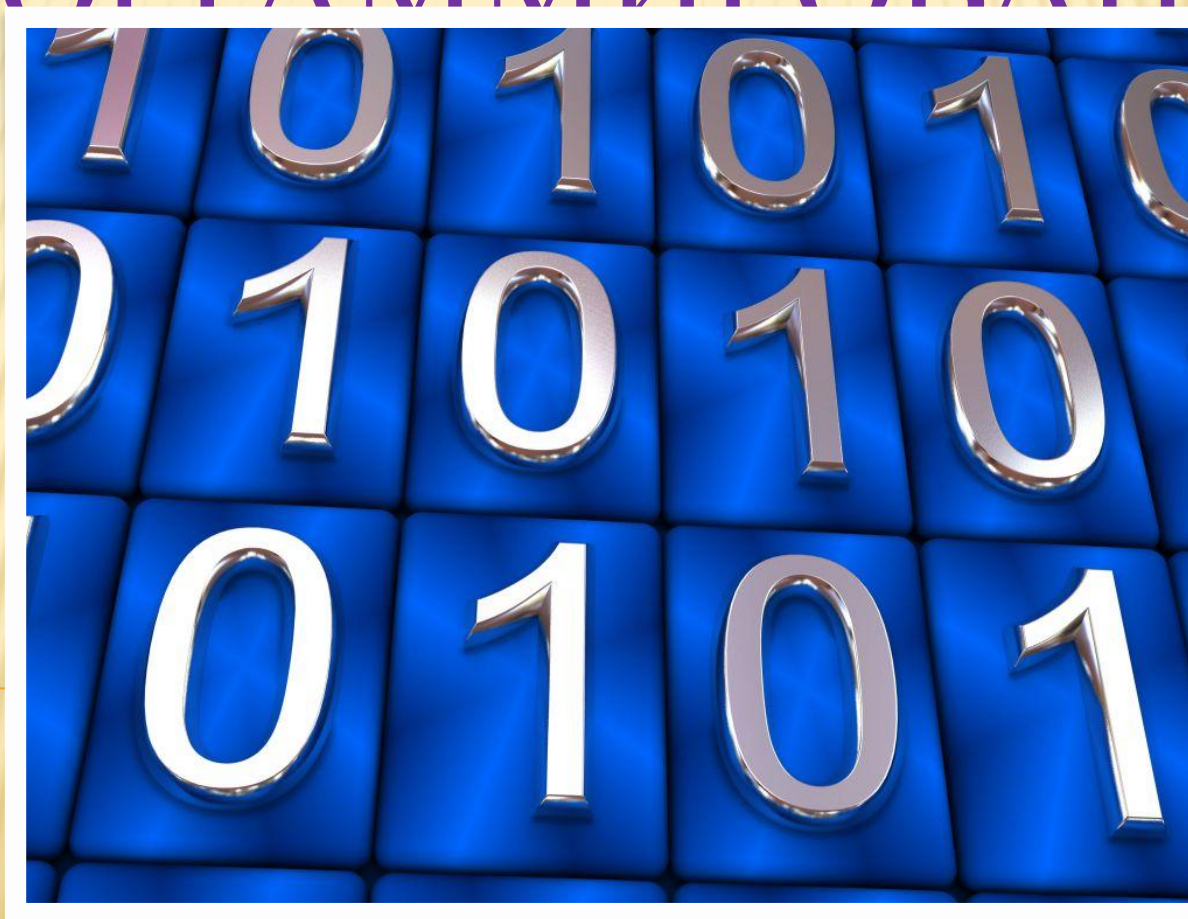
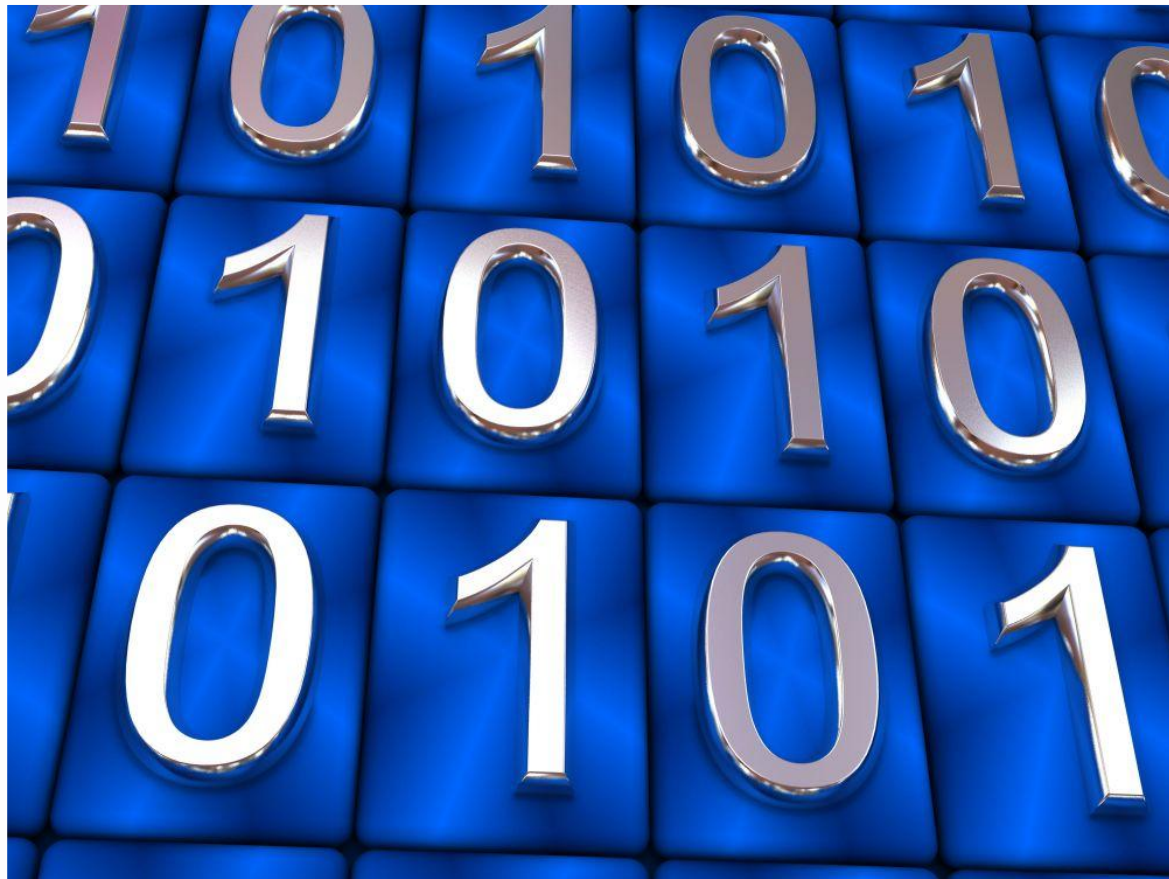


ЯЗЫКИ ПРОГРАММИРОВАНИЯ

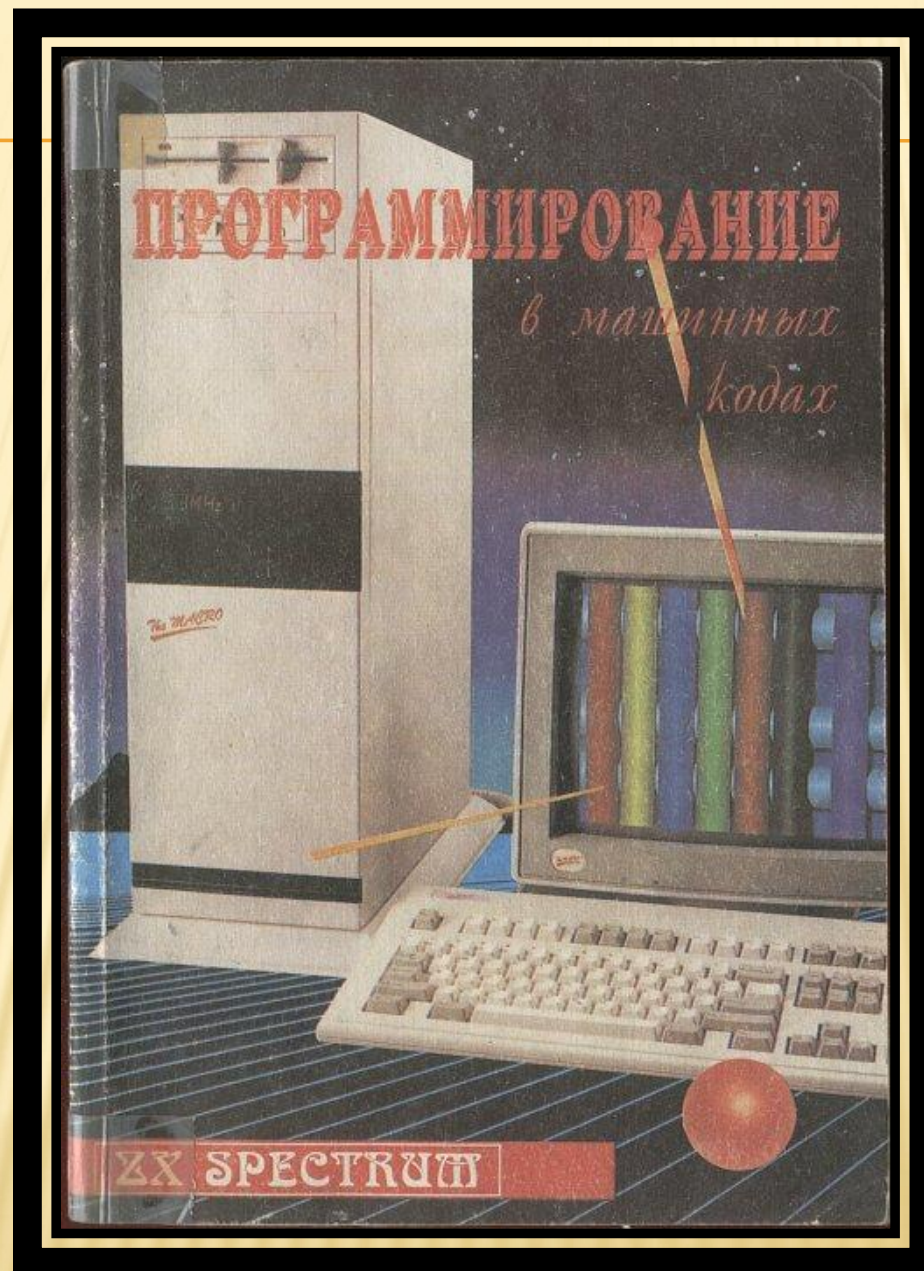


- Язык программирования — формальная знаковая система, предназначенная для записи компьютерных программ. Язык программирования определяет набор лексических элементов и правил их комбинации.



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				ω-1	t мксек	Операция	КОП	Результат	ω-1	t мксек	Операция	КОП	Величина сдвига	ω-1	t мксек	КОП	Код в РА	Передача управл в A ₂	Передача управл в КРА+1	t мксек					
	0xH	60	ВH	В0xH																		КОП	Код в РА	Передача управл в A ₂	Передача управл в КРА+1	
Сложение	01	21	41	61	Z < 0	285	Сравнение	15	C _к = F _к - Q _к	C = 0	24	Сдвиг кода по A ₁	54	C _к = Q _к - P _к ; S = P - 100	C = 0	24x55	11	РА < A ₁ и ω = 1	РА > A ₁ или ω = 0	24						
Вычитание	02	22	42	62			Сравнение с ост	35	аналогично 15, но по DCT при C ≠ 0			Сдвиг кода по P	74	C _к = Q _к ; S = P - 100			C = 0	24x55	31		РА > A ₁ и ω = 1	РА < A ₁ или ω = 0				
Вычитание модулей	03	23	43	63			Логическое умножение	55	C _к = F _к ∧ Q _к			Сдвиг мантиссы по P	14	C _к = Q _к ; S = A ₁ ¹⁰⁰ - 100			C = 0	24x55	51		РА < A ₁ и ω = 0	РА > A ₁ или ω = 1				
Деление	04	24	-	-			Логическое сложение	75	C _к = F _к ∨ Q _к			Сдвиг мантиссы по P	34	C _к = Q _к ; S = P - 100			C = 0	24x55	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					24									
Извлечение кв. корня	44	64	-	-			Цикл сложения	07	C _к = (F _к + 1) mod 2 ⁹	24	Операция	КОП	Результат	ω-1	t мксек	12		A ₃	17	Засылка кода [Z] → Y	у-ячейка по A ₂	32	РА > A ₁	РА < A ₁		
Выход младших разрядов произведения	47						Цикл вычитания	27	C _к = (F _к - 1) mod 2 ⁹		Выборка из динамики	17	Z → Y	у-ячейка по A ₂	C = 0	90			40	РА < зона A ₂ [A ₁]	РА > зона A ₂ [A ₁]	60	РА > A ₁	РА < A ₁		
Операции изменения порядка							Сложение мантисс	13	C _к = F _к		Засылка в динамику	37	Засылка кода [Y] → Z	у-ячейка по A ₂	24	24			16	Команды передачи управления с занесением в A ₃						
Операция	КОП	Результат	ω-1	t мксек	Вычитание мантисс	33	C _к = F _к	Засылка кода	00		[A ₁] → A ₂	A ₂ безразличен	24	24			36			КОД заносимый в A ₃						
Сложение порядка с A ₁	06	Γ = q + (A ₁ ¹⁰⁰ - 100)	Γ < 100	24	Сложение порядка с P	26	Γ = q + (P - 100)	Засылка кода с КЗУ	20	[КЗУ] → A ₃	Эта запись по A ₂ ; N _к - 3	24						24		76	КОД заносимый в A ₃					
Сложение порядка с P	26	Γ = q + (P - 100)			Вычитание порядка из A ₁	46	Γ = q - (A ₁ ¹⁰⁰ - 100)	Сложение КОПов	53	C _к = (F _к + 1) mod 2 ⁹	20										A ₁ = 0000 - переход на 0 Траб АМ	24	24	76	КОД заносимый в A ₃	
Вычитание порядка из A ₁	46	Γ = q - (A ₁ ¹⁰⁰ - 100)			Вычитание КОПов	73	C _к = (F _к - 1) mod 2 ⁹	Вычитание КОПов	73	C _к = F _к	20				A ₁ = 0000 - переход на 0 Траб АМ	24			24		76				КОД заносимый в A ₃	
Вычитание порядка из P	66	Γ = q - (P - 100)			Цикл сдвиг	67	C _к = F _к mod 2 ⁹	Цикл сдвиг	67	C _к = F _к	20		A ₁ = 0000 - переход на 0 Траб АМ	24	24		76								КОД заносимый в A ₃	
Команда останова					Операции в M2, выполняемые по сигналам из M1					Операции, выполняемые по командам Ma и Mb								24								
КОП	Совмещенные операции				Операция					Режим										24						
77	Вызов [A ₁] и [A ₂] на P, и A ₂ гашение [A ₃] при проболжени				Выборка команд в M2 по адресу из M1					УЧ						24										
Операции изменения [РА]					Обмен кодами M1 ↔ M2					Обмен с МБ					24											
КОП	Код в РА	Код в A ₃	t мксек		AP1 _{M2}					Обмен с МЛ							24									
52	A ₂	0520000 A ₁ 0000	24		AP1 _{M2}					Печать „8“								24								
72	зона A ₂ [A ₂]	0520000 A ₁ 0000	24		AP1 _{M2}					Печать „10“						24										
Команды ввода					Операции выполняемые по сигналам от линии связи и аналоговой машины					Печать Ял.Ц					24											
Операция	КОП	A ₁	A ₂	A ₃	AP1 _{M2}					Перфорация							24									
Ввод с остановом при несоблюдении КΣ	10	A _н МЗУ	КРА	AKΣ	AP1 _{M2}					Накопление								24								
Ввод без ост. при несоблюд КΣ	30	A _н МЗУ	КРА	AKΣ	AP1 _{M2}					Разметка МП						24										
Операция изменения регистров приращения					Операция					Подвод МЛ					24											
КОП	A ₁	A ₂	код заносимый в A ₃	t мксек	Авторазрыв от ЛС					Возврат состояния							24									
57	2,1p-PPM1	3,2p-PPM2	3,3p-PPM3	24	Авторазрыв от АМ					Возврат состояния								24								
54p-PPME	3,2p-PPA3	6,3p-PPA2	8,7p-PPA1	24	Авторазрыв от АМ					Возврат состояния						24										
7p-Pr A3	6,3p-PPA2	8,7p-PPA1	12,1p-PPA1	24	Авторазрыв от АМ					Возврат состояния					24											
8p-Pr A2	8,7p-PPA1	12,1p-PPA1	16,5p-PPA1	24	Авторазрыв от АМ					Возврат состояния							24									
9p-Pr A1	12,1p-PPA1	16,5p-PPA1	20,9p-PPA1	24	Авторазрыв от АМ					Возврат состояния								24								
10p-Pr A2	16,5p-PPA1	20,9p-PPA1	25,3p-PPA1	24	Авторазрыв от АМ					Возврат состояния						24										
11p-Pr M1	20,9p-PPA1	25,3p-PPA1	30,7p-PPA1	24	Авторазрыв от АМ					Возврат состояния					24											
12p-Pr M5	25,3p-PPA1	30,7p-PPA1	35,1p-PPA1	24	Авторазрыв от АМ					Возврат состояния							24									

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 МЛ.

- ***
1. При чтении адреса с ПК 1^й р. РБФ-блокировка АВ ост при несоблюдении КΣ
 - 1^й 13p РБФ-блокировка записи в МЗУ последующего массива 37-38p РБФ(РПА) → → 13-14p СМЯ

2. При A₁ = 0 блокируется запись в МЗУ последующего массива.
- * Обращение к МЗУ по команде ИРП производится по старым значениям ИРП МЗУ. РП изменяется только по командам ИРП, АРЗВ.
- * Операции АР1_{M2}, АР2_{M2}, АРЛС, 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; [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

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!)