

**Язык
С**



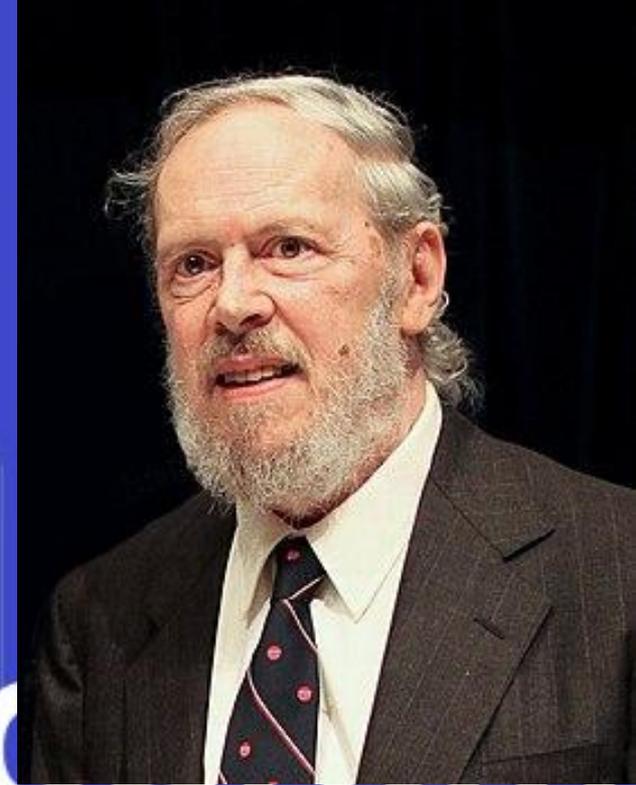
Program

Канарейкин А. И.

История

Си (англ. C) — компилируемый статически типизированный язык программирования общего назначения, разработанный в 1969—1973 годах сотрудником Bell Labs Деннисом Ритчи как развитие языка Би.

Первоначально был разработан для реализации операционной системы UNIX, но впоследствии был перенесён на множество других платформ. Согласно дизайну языка, его конструкции близко сопоставляются типичным машинным инструкциям, благодаря чему он нашёл применение в проектах, для которых был свойственен язык ассемблера, в том числе как в операционных системах, так и в различном прикладном программном обеспечении для множества устройств — от суперкомпьютеров до встраиваемых систем. Язык программирования Си оказал существенное влияние на развитие индустрии программного обеспечения, а его синтаксис стал основой для таких языков программирования, как C++, C#, Java и Objective-C.



Основное отличие Unix-подобных систем от других операционных систем заключается в том, что это изначально многопользовательские многозадачные системы. В Unix может одновременно работать сразу много людей, каждый за своим терминалом, при этом каждый из них может выполнять множество различных вычислительных процессов, которые будут использовать ресурсы именно этого компьютера.

Вторая колоссальная заслуга Unix — в её мультиплатформенности. Ядро системы разработано таким образом, что его легко можно приспособить практически под любой микропроцессор.

Unix имеет и другие характерные особенности:

использование простых текстовых файлов для настройки и управления системой;

широкое применение утилит, запускаемых из командной строки;

взаимодействие с пользователем посредством виртуального устройства — терминала;

представление физических и виртуальных устройств и некоторых средств

межпроцессового взаимодействия в виде файлов;

использование конвейеров из нескольких программ, каждая из которых выполняет одну задачу.

Применение

В настоящее время Unix-системы распространены в основном среди серверов, а также как встроенные системы для различного оборудования, включая смартфоны. Также Unix-системы доминируют на суперкомпьютерах, в частности, на 100 % суперкомпьютеров из рейтинга TOP500 установлена ОС Linux.

Среди ОС для рабочих станций и домашнего применения Unix и Unix-подобные ОС занимают после Microsoft Windows второе (macOS), третье (GNU/Linux) и многие последующие места по популярности.

Основные особенности Си:

Простая языковая база, из которой в стандартную библиотеку вынесены многие существенные возможности, вроде математических функций или функций работы с файлами; ориентация на процедурное программирование; система типов, предохраняющая от бессмысленных операций; использование препроцессора для абстрагирования однотипных операций; доступ к памяти через использование указателей; небольшое число ключевых слов;

В то же время в Си

отсутствуют:

Вложенные функции; Прямое возвращение нескольких значений из функций; Сопрограммы; Средства автоматического управления памятью; Строенные средства объектно-ориентированного программирования; Средства функционального программирования.

Program

Основные понятия языка

Программа, написанная на языке Си, состоит из операторов. Каждый оператор вызывает выполнение некоторых действий на соответствующем шаге выполнения программы.

При написании операторов применяются латинские прописные и строчные буквы, цифры и специальные знаки. К таким знакам, например, относятся: точка (.), запятая (,), двоеточие (:), точка с запятой (;) и др. Совокупность символов, используемых в языке, называется алфавитом языка.

В персональном компьютере символы хранятся в виде кодов. Соответствие между каждым символом и его кодом задается специальной кодовой таблицей. На нее разработан стандарт ASCII, поэтому коды символов называют ASCII-кодами.

Различают *видимые* и *управляющие символы*. Первые могут быть отображены на экране дисплея либо отпечатаны на принтере. Вторые вызывают определенные действия в машине, например: звуковой сигнал - код 7_{10} , возврат курсора на один шаг - код 8_{10} , горизонтальная табуляция - код 9_{10} , перевод курсора на новую строку - код 10_{10} , перемещение курсора в начало строки - код 13_{10} и т.д. Такие управляющие символы имеют десятичные номера 0 - 31, 127.

Для представления каждого символа в персональном компьютере используется один байт, поэтому общее число символов равно $2^8 = 256$. Кодовая таблица, которая устанавливает соответствие между символом и его кодом, имеет 256 строк вида:

код_символа_в_заданной_системе_счисления - символ.

Первая половина кодовой таблицы является стандартной, а вторая используется для представления символов национальных алфавитов, псевдографических

Важным понятием языка является идентификатор, который используется в качестве имени объекта (функции, переменной, константы и др.). Идентификаторы должны выбираться с учетом следующих правил:

Они должны начинаться с буквы латинского алфавита (a,...,z, A,...,Z) или с символа подчеркивания (_).

В них могут использоваться буквы латинского алфавита, символ подчеркивания и цифры (0,...,9). Использование других символов в идентификаторах запрещено.

В языке Си буквы нижнего регистра (a,...,z), применяемые в идентификаторах, отличаются от букв верхнего регистра (A,...,Z). Это означает, что следующие идентификаторы считаются разными: name, NaMe, NAME и т.д.

Идентификаторы могут иметь любую длину, но воспринимается и используется для различения объектов (функций, переменных, констант и т.д.) только часть символов. Их число меняется для разных систем программирования, но в соответствии со стандартом ANSI C не превышает 32 (в Си++ это ограничение снято). Если длина идентификатора установлена равной 5, то имена count и counter будут идентичны, поскольку у них совпадают первые пять символов.

Идентификаторы для новых объектов не должны совпадать с ключевыми словами языка и именами стандартных функций из библиотеки.

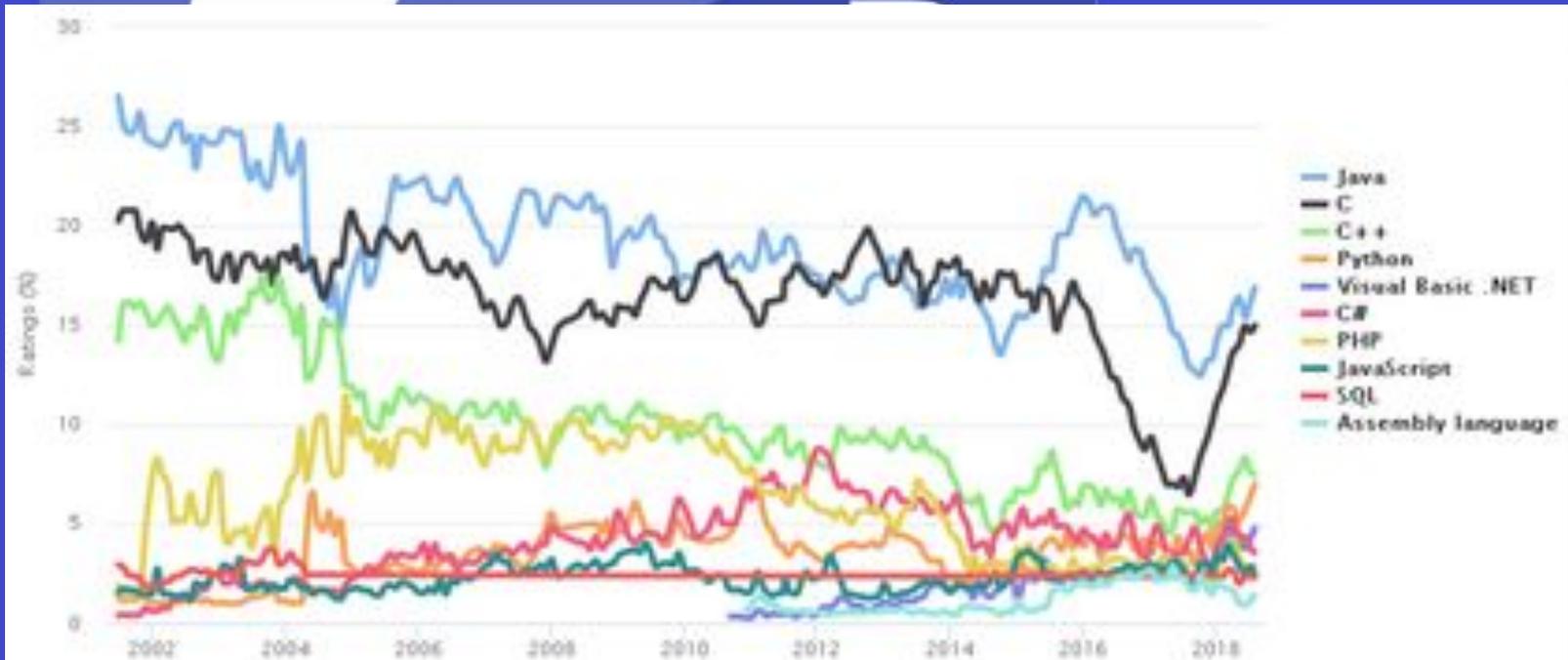
В программах на языке Си важная роль отводится комментариям. Они повышают наглядность и удобство чтения программ. Комментарии обрамляются символами /* и */. Их можно записывать в любом месте программы.

В языке Си++ введена еще одна форма записи комментариев. Все, что находится после знака // до конца текущей строки, будет также рассматриваться как комментарий. Отметим, что компилятор языка Си, встроенный в систему программирования Borland C++, позволяет использовать данный комментарий и в программах на Си.

Пробелы, символы табуляции и перехода на новую строку в программах на Си игнорируются. Это позволяет записывать различные выражения в хорошо читаемом виде. Кроме того, строки программы можно начинать с любой позиции, что дает возможность выделять в тексте группы

Область применения

Язык широко применяется при разработке операционных систем, на уровне прикладного интерфейса операционных систем, во встраиваемых системах, а также для создания высокопроизводительного или критического в плане обработки ошибок кода. Одной из причин широкого распространения для программирования на низком уровне является возможность писать кроссплатформенный код, который может по-разному обрабатываться на разном оборудовании и на разных операционных системах.



m

Проблемы и критика

Язык весьма сложен и наполнен опасными элементами, которые очень легко использовать неправильно. Своей структурой и правилами он никак не поддерживает программирование, нацеленное на создание надёжного и удобного в сопровождении программного кода, напротив, рождённый в эпоху прямого программирования под различные процессоры, язык способствует написанию небезопасного и запутанного кода. Многие профессиональные программисты склонны считать, что язык Си — мощный инструмент для создания элегантных программ, но в то же время с его помощью можно создавать крайне некачественные решения.

Из-за различных допущений в языке программы могут компилироваться с множественными ошибками, что часто приводит к непредсказуемому поведению программы. Современные компиляторы предоставляют опции для статического анализа кода, но даже они не способны выявить все возможные ошибки.

Результатом неграмотного программирования на Си могут стать уязвимости программного обеспечения, что может сказаться на безопасности его использования.