Федеральное государственное бюджетное образовательное учреждение высшего образования «Ульяновский государственный университет»

Кафедра ……

РЕФЕРАТ

на тему:

«История языка СИ»

Выполнил :

студент 1 курса группы …

Яшин. К.П.

Проверил:

Ульяновск

2019

**ОГЛАВЛЕНИЕ**

Введение

1. История возникновения

2. Знакомство с языком СИ

3. Особенности языка, его достоинства и недостатки

Заключение

**ВВЕДЕНИЕ**

Компьютеры достаточно давно и прочно вошли в нашу жизнь. Они кардинально поменяли мир и возможности людей. Компьютер гораздо облегчил нашу жизнь. Иногда мы уже не представляем себе нашу жизнь без компьютера и интернета. Еще несколько лет назад было редкостью увидеть какой-нибудь персональный компьютер – они были, но были очень дорогие, и даже не каждая фирма могла иметь у себя в офисе компьютер. А теперь? Теперь в каждом третьем доме есть компьютер, который уже глубоко вошел в жизнь человека.

Современные вычислительные машины представляют одно из самых значительных достижений человеческой мысли, влияние, которого на развитие научно-технического прогресса трудно переоценить. Области применения ЭВМ непрерывно расширяются. По своему назначению компьютер – универсальное техническое устройство для работы с информацией. По принципам устройства компьютер – модель человека, работающего с информацией. Языки программирования этой эры, как и компьютеры на которых они использовались, были разработаны для специфичных задач, таких как научные вычисления. Поскольку машины были дорогими и лишь одна задача выполнялась за раз, то и машинное время было дорого – поэтому скорость выполнения программы стояла на первом месте.

На современном этапе развития компьютерных технологий невозможно представить какого–либо высококвалифицированного специалиста, не владеющего информационными технологиями. Поскольку деятельность любого субъекта в значительной степени зависит от степени владения информации, а также способности эффективно ее использовать. Для свободной ориентации в информационных потоках современный специалист любого профиля должен уметь получать, обрабатывать и использовать информацию, прежде всего, с помощью компьютеров, а также телекоммуникаций и других новейших средств связи, в том числе и уметь, обращаться с языками программирования.

В этих условиях - когда мощность стала дешева и доступна - создатели языков программирования все больше стали задумываться об удобстве написания программ, а не только скорости их выполнения. «Мелкие» (атомарные) операции, выполняемые непосредственно устройствами машины, объединили в более «крупные», высокоуровневые операции и целые конструкции, с которыми человеку куда проще и удобнее работать.

Актуальность данной темы обусловлена тем, что прогресс компьютерных технологий определил процесс появления новых разнообразных знаковых систем для записи алгоритмов – языков программирования.

**1.История возникновения**

СИ – это "сопутствующий продукт", полученный во время создания операционной системы UNIX, которая разрабатывалась в Bell Laboratories Кеном Томпсоном, Денисом Ритчи и ко. Томпсон в одиночку написал оригинальную версию UNIX, которая работала на DEC PDP-7, одном из первых миникомпьютеров всего с 8К слов в основной памяти Язык программирования Си разрабатывался в период с 1969 по 1973 годы в лабораториях Bell Labs, и к 1973 году на него была переписана большая часть ядра UNIX, первоначально написанного на ассемблере PDP-11/20. Название языка стало логическим продолжением старого языка «Би»[a], многие особенности которого были положены в основу.

Как и остальные операционные системы того времени, UNIX был написан на ассемблере. Отладка программ на ассемблере настоящая мука и их проблематично улучшать, и UNIX не был исключением. Томпсон решил, что для дальнейшей разработки ОС необходим язык высокого уровня и придумал небольшой язык B. За основу Томпсон взял язык BCPL, язык для системного программирования, разработанный в середине 60-х. BCPL, в свою очередь, берёт начало от Алгола 60, одного из самых первых (и оказавших наибольшее влияние) языков.

Ритчи вскоре присоединился к проекту UNIX и начал писать на B. В 1970 Bell Labs приобрела для проекта PDP-11. Так как B был готов к работе на PDP-11, Томпсон переписал часть UNIX на B. В 1971 стало ясно, что B не совсем подходит для PDP-11, поэтому Ритчи начал создавать расширенную версию B. Сначала он назвал её NB (New B), но когда язык стал сильно отличаться от B, название сменили на C. Язык к 1973 стал достаточно стабилен для того, чтобы на нём можно было переписать UNIX. Переход на C обеспечил важное преимущество: переносимость. Написав компилятор C для каждой из машин в Bell Labs, команда разработчиков могла портировать на них UNIX.

СИ продолжил развиваться в 70-х, особенно в период с 1977 по 1979, когда вышла первая книга по C. Книга "Язык программирования Си", написанная Брайаном Керниганом и Денисом Ритчи и опубликованная в 1978 стала библией программистов на Си. При отсутсвии официального стандарта эта книга – известная также как K&R, или "Белая Книга", как любят называть поклонники си – фактически стала стандартом. В 70-х программистов на C было немного и большинство из низ были пользователями UNIX. Тем не менее, в 80-х C вышел за узкие рамки мира UNIX. Компиляторы C стали доступны на различных машинах, работающих под управлением разных операционных систем. В частности, Си стал распространяться на быстро развивающейся платформе IBM PC.

Вместе с ростом популярности появились проблемы. Программисты, писавшие новые компиляторы брали за основу язык, описанный в K&R. К сожалению, в K&R некоторые особенности языка были описаны расплывчато, поэтому компиляторы часто трактовали их на своё усмотрение. Кроме того, в книге не было чёткого разделения между тем, что является особенностью языка, а что особенностью операционной системы UNIX. Ухудшало ситуацию и то, что после публикации K&R Си продолжал развиваться: в него добавлялись новые возможности и из него вырезались старые. Вскоре появилась очевидная необходимость в исчерпывающем, точном и соответствующем современным требованиям описании языка. Без такого стандарта стали появляться диалекты языка, которые мешали переносимости – сильнейшей стороне языка.

Разработка американского стандарта Си началась в 1983 под покровительством Американского Национального Института Стандартов (ANSI). После многих доработок стандарт был закончен в 1988 и формально принят в декабре 1989 как стандарт ANSI X3.159-1989. В 1990 году он был утверждён международной организацией по стандартизации (ISO) как интернациональный стандарт ISO/IEC 9899:1990. Эту версию языка обычно называют C89 или C90, для того, чтобы не путать её с оригинальной версией Си, которую обычно называют K&R C.

Язык подвергся небольшим изменениям в 1995 (изменения описаны в документе, который обычно называют Поправка 1). Более значительные изменения случились в 1999 году, когда был опубликован стандарт ISO/IEC 9899:1999. Язык, описанный в этом стандарте обычно называют C99. Термины "ANSI C", "ANSI/ISO C" и "ISO C", когда-то используемые для описания C99 из-за существования двух стандартов имеют двоякое толкование.

В 2011 году вместе с редакцией языка Си++ был выпущен стандарт C11. Несмотря на наличие стандарта 11 года, многие компиляторы до сих пор не поддерживают полностью даже версии C99, так что использование стандарта С11 будет указано явно.

Лахатин, А.С. Языки программирования. Учеб. пособие / А.С. Лахатин, Л.Ю. Искакова. - Екатеринбург, 1998. - 548с.: ил.

Язык программирования Си = The C programming language. — 2-е изд. — М.: Вильямс, 2007. — С. 304.

**2. Знакомство с языком СИ**

Язык программирования СИ отличается минимализмом. Авторы языка хотели, чтобы программы на нём легко компилировались с помощью однопроходного компилятора, чтобы каждой элементарной составляющей программы после компиляции соответствовало весьма небольшое число машинных команд, а использование базовых элементов языка не задействовало библиотеку времени выполнения. Однопроходный компилятор компилирует программу, не возвращаясь назад, к уже обработанному тексту. Поэтому использованию функции должно предшествовать её объявление. Код на СИ можно легко писать на низком уровне абстракции, почти как на ассемблере.

Иногда СИ называют «универсальным ассемблером» или «ассемблером высокого уровня», что отражает различие языков ассемблера для разных платформ и единство стандарта Си, код которого может быть скомпилирован без изменений практически на любой модели компьютера. Си часто называют языком среднего уровня или даже низкого уровня, учитывая то, как близко он работает к реальным устройствам.

Компиляторы СИ разрабатываются сравнительно легко благодаря относительно низкому уровню языка и скромному набору элементов. Поэтому данный язык доступен на самых различных платформах (возможно, круг этих платформ шире, чем у любого другого существующего языка). К тому же, несмотря на свою низкоуровневую природу, язык позволяет создавать переносимые программы и поддерживает в этом программиста. Программы, соответствующие стандарту языка, могут компилироваться на самых различных компьютерах.

СИ (как и ОС UNIX, с которой он долгое время был связан) создавался программистами и для программистов, круг которых был бы ненамного шире круга разработчиков языка. Несмотря на это, область использования языка значительно шире задач системного программирования.

СИ создавался с одной важной целью: сделать более простым написание больших программ с минимумом ошибок по правилам процедурного программирования, не добавляя на итоговый код программ лишних накладных расходов для компилятора, как это всегда делают языки очень высокого уровня, такие как Бейсик. С этой стороны Си имеет следующие важные особенности:

- простую языковую базу, из которой вынесены в библиотеки многие существенные возможности, вроде математических функций или функций управления файлами;

-ориентацию на процедурное программирование, обеспечивающую удобство применения структурного стиля программирования;

-систему типов, предохраняющую от бессмысленных операций;

-использование препроцессора для, например, определения макросов и включения файлов с исходным кодом;

-непосредственный доступ к памяти компьютера через использование указателей; минимальное число ключевых слов;

-передачу параметров в функцию по значению, а не по ссылке (при этом передача по ссылке выполняется с помощью указателей);

-указатели на функции и статические переменные области действия имён;

-записи — определяемые пользователем собирательные типы данных (структуры), которыми можно манипулировать как одним целым.

При всём вышесказанном, существуют некоторые особенности других языков, которыми язык СИ не наделён:

- автоматическое управление памятью;

-поддержка объектно-ориентированного программирования (при этом первые версии C++ генерировали код программы на языке Си);

-замыкание;

-вложенные функции (существуют компиляторы языка Си реализующие эту функцию, например, компилятор GNU);

-полиморфизм функций и операторов;

-встроенная поддержка многозадачности и сети функции высшего порядка карринг.

После появления язык Си был хорошо принят, потому что он позволял быстро создавать компиляторы для новых платформ, а также позволял программистам довольно точно представлять, как выполняются их программы. Благодаря этому программы, написанные на Си, эффективнее написанных на многих других языках. Как правило, лишь оптимизированный вручную код на ассемблере может работать ещё быстрее, потому что он даёт полный контроль над машиной, однако развитие современных компиляторов вместе с усложнением современных процессоров быстро сократило этот разрыв.Богатырев, А. Язык программирования С [Электронный ресурс] / А. Богатырев.- электр. дан. - Режим доступа: http://www.refby.com. - Загл. с экрана.

**3. Особенности языка, его достоинства и недостатки**

Авторы CИ стремились создать язык, сочетающий простоту и выразительность современных объектно-ориентированных языков (вроде Java) c богатством возможностей и мощью C++. Наиболее важная новая характеристика СИ, по сравнению с предшествующими языками, - это поддержка возможности взаимодействия нескольких компьютерных языков (многоязыкового программирования). По словам Андерса Хейлсберга, CИ позаимствовал большинство своих синтаксических конструкций из C++. В частности, в нем присутствуют такие удобные типы данных, как структуры и перечисления (другой потомок C++ -- Java -- лишен этих элементов, что создает определенные неудобства при программировании). Синтаксические конструкции СИ унаследованы не только от C++, но и от C, и от Visual Basic. Например, в СИ, как и в Visual Basic, используются свойства классов. Как C++, СИ позволяет производить перегрузку операторов для созданных вами типов.

Различают несколько типов данных в работе с языком СИ, познакомимся с некоторыми из них:

-*Примитивные* (Поддерживает идею примитивных типов (которые в CИ являются подмножеством типов-значений — value types);

-*Ссылочные типы* (Переменные ссылочных типов, называемые объектами, сохраняют ссылки на фактические данные. Для объявления ссылочных типов используются следующие ключевые слова: class, interface, delegate. Также существуют встроенные типы: object и string. Интерфейс содержит только подписи методов, свойств, событий или индексаторов);

-*Структуры* (CИ позволяет создавать пользовательские типы-значения, используя ключевое слово struct. В отличие от экземпляров классов, экземпляры типов-значений создаются не в куче, а на стеке вызовов или в составе экземпляра объекта, в котором они объявлены, что в некоторых случаях повышает производительность кода. С точки зрения программиста они подобны классам, но с несколькими ограничениями);

-*Перечислимые типы (п*еречислимые типы в CИ происходят от примитивных целочисленных типов. Допустимым значением перечислимого типа является любое значение лежащего в его основе примитивного, хотя для его присваивания может потребоваться явное приведение типа);

*- Массивы и коллекции* (массивы и коллекции получили выражение в синтаксисе, благодаря особой разновидности цикла for (цикл по коллекции, известный также как цикл foreach). Массив является объектом класса Array);

-И так далее.

Язык СИ имеет свои существенные особенности, давайте перечислим некоторые из них:

-СИ обеспечивает полный набор операторов структурного программирования. СИ предлагает необычно большой набор операций. Многие операции СИ соответствуют машинным командам, и поэтому допускают прямую трансляцию в машинный код. Разнообразие операций позволяет выбирать их различные наборы для минимизации результирующего кода.

-СИ поддерживает указатели на переменные и функции. Указатель на объект программы соответствует машинному адресу этого объекта. Посредством разумного использования указателей можно создавать эффективно-выполняемые программы, так как указатели позволяют ссылаться на объекты тем же самым путем, как это делает машина. СИ поддерживает арифметику указателей, и тем самым позволяет осуществлять непосредственный доступ и манипуляции с адресами памяти.

-В своем составе СИ содержит препроцессор, который обрабатывает текстовые файлы перед компиляцией. Среди его наиболее полезных приложений при написании программ на СИ являются: определение программных констант, замена вызовов функций аналогичными, но более быстрыми макросами, условная компиляция. Препроцессор не ограничен процессированием только исходных текстовых файлов СИ, он может быть использован для любого текстового файла.

-СИ-гибкий язык, позволяющий принимать в конкретных ситуациях самые разные решения. Тем не менее, СИ налагает незначительные ограничения в таких, например, действиях, как преобразование типов. Во многих случаях это является достоинством, однако программисты должны хорошо знать язык, чтобы понимать, как будут выполняться их программы.

История языка СИ неразрывно связана с историей операционной системы UNIX. Эта система, как и большинство входящих в нее утилит, написана на СИ. С тех пор СИ остается широко распространенным языком — по тем же причинам, что и Фортран: с расширением круга пользователей язык развивает сокрушительную инерцию. Именно по этой причине этот язык получил свою популярность. Какие же достоинства у этого языка? Их не мало:

1. СИ современный язык. Он включает в себя те управляющие конструкции, которые рекомендуются теоретическим и практическим программированием. Результатом такого подхода является надежная и читаемая программа.

2. СИ — эффективный язык. Его структура позволяет использовать возможности современных ЭВМ. На языке СИ программы обычно отличаются компактностью и быстротой исполнения.

3. СИ — переносимый, или мобильный, язык. Если модификации все-таки необходимы, то часто они могут быть сделаны путем простого изменения нескольких элементов в “головном” файле, который сопутствует главной программе. Он предоставляет исключительные возможности для переноса программ. Компиляторы с данного языка реализованы почти на 40 типах вычислительных систем, начиная от- 8-разрядных микропроцессоров и кончая CRAY-1 одним из самых мощных в настоящее время суперкомпьютеров.

4. СИ — мощный и гибкий язык. Программы, написанные на Си, используются для решения физических и технических проблем и даже для производства мультипликационных фильмов.

5. СИ обладает рядом замечательных конструкций управления, обычно ассоциируемых с ассемблером.

6. СИ — удобный язык. Он достаточно структурирован, чтобы поддерживать хороший стиль программирования, и вместе с тем не связывать вас с ограничениями.

Также, как и особенностей, в языке Си присутствует куча недостатков. Ведь от них не защищен не один проект, в том числе проект разработки и выполнения программ, на языке СИ. Язык Си предъявляет достаточно высокие требования к квалификации использующего его программиста. При изучении СИ желательно иметь представление о структуре и работе компьютера. Большую помощь и более глубокое понимание идей СИ, как языка системного программирования, обеспечат хотя бы минимальное знание языка ассемблер. Уровень старшинства некоторых операторов не является общепринятым, некоторые синтаксические конструкции могли бы быть лучше. Тем не менее, как оказалось СИ – чрезвычайно эффективный и выразительный язык, пригодный для широкого класса задач.

Уэйт, М. Язык С. Руководство для начинающих. / М. Уэйт, С. Прага, Д. Мартин. - М.: Мир, 1995. - 521с.: ил.

Программирование на языке Си = Programming in C. — 3-е изд. — М.: Вильямс, 2006. — С. 496. ISBN -3

Язык программирования Си для «чайников» = C For Dummies. — М.: Диалектика, 2006. — С. 352. — ISBN -4

**ЗАКЛЮЧЕНИЕ**

Ввиду очень удобного объектно-ориентированного дизайна, Cи является хорошим выбором для быстрого конструирования различных компонентов - от высокоуровневой бизнес логики до системных приложений, использующих низкоуровневый код.

Также следует отметить, что CИ является и Web ориентированным - используя простые встроенные конструкции языка ваши компоненты могут быть легко превращены в Web сервисы, к которым можно будет обращаться из Internet посредством любого языка на любой операционной системе.

Дополнительные возможности и преимущества перед другими языками приносит в CИ использование передовых Web технологий, таких как: XML (Extensible Markup Language) и SOAP (Simple Object Access Protocol). Среда разработки Web сервисов позволяет программисту смотреть на существующие сегодня Web приложения, как на родные CИ объекты, что дает возможность разработчикам соотнести имеющиеся Web сервисы с их познаниями в объектно-ориентированном программировании.

Последнее время С и С++ являются наиболее используемыми языками для разработки коммерческих и бизнес приложений. Сейчас существуют языки, увеличивающие продуктивность разработки за счет потери в гибкости, которая так привычна и необходима программистам на СИ. Учитывая все подобные пожелания, Microsoft разработала новый язык - CИ. В него входит много полезных особенностей - простота, объектная ориентированность, типовая защищенность, "сборка мусора", поддержка совместимости версий и многое другое. Данные возможности позволяют быстро и легко разрабатывать приложения, особенно COM+ приложения и Web сервисы. При создании CИ, его авторы учитывали достижения многих других языков программирования.

В CИ была унифицирована система типов, теперь вы можете рассматривать каждый тип как объект. Несмотря на то, используете вы класс, структуру, массив или встроенный тип, вы можете обращаться к нему как к объекту. Объекты собраны в пространства имен (namespaces), которые позволяют программно обращаться к чему-либо. Это значит, что вместо списка включаемых файлов заголовков в своей программе вы должны написать какие пространства имен, для доступа к объектам и классам внутри них, вы хотите использовать.

В наше время, когда усиливается связь между миром коммерции и миром разработки программного обеспечения, и корпорации тратят много усилий на планирование бизнеса, ощущается необходимость в соответствии абстрактных бизнес процессов их программным реализациям. К сожалению, большинство языков реально не имеют прямого пути для связи бизнес логики и кода.

Например, сегодня многие программисты комментируют свои программы для объяснения того, какие классы реализуют какой-либо абстрактный бизнес объект. CИ позволяет использовать типизированные, расширяемые метаданные, которые могут быть прикреплены к объекту. Архитектурой проекта могут определяться локальные атрибуты, которые будут связанны с любыми элементами языка - классами, интерфейсами и т.д. Разработчик может программно проверить атрибуты какого-либо элемента. Это существенно упрощает работу, к примеру, вместо того чтобы писать автоматизированный инструмент, который будет проверять каждый класс или интерфейс, на то, является ли он действительно частью абстрактного бизнес объекта, можно просто воспользоваться сообщениями основанными на определенных в объекте локальных атрибутах.