**Государственное бюджетное профессиональное образовательное учреждение  
«Нижегородский Губернский Колледж»**

Методическая комиссия «Информатика и вычислительная техника»

Допущен к защите:

Преподаватель

\_\_\_\_\_\_\_\_Мухин Н.А.

« » декабрь 2019 г.

**ОТЧЕТ ПО УЧЕБНОЙ ПРАКТИКЕ**

ПМ 03 «Сопровождение и продвижение ПО отраслевой направленности»

Руководитель: \_\_\_\_\_\_\_\_ Мухин Н.А.

Студент: \_\_\_\_\_\_\_\_\_\_\_\_ Карагулин Д.А.

Специальность, группа: 09.02.05, 41Пр

Нижний Новгород  
2019 г.

**Содержание**

[Введение 3](#_Toc26543866)

[**Теоретическая часть** 4](#_Toc26543867)

[**Объект прохождения практики** 4](#_Toc26543868)

[Выполнение сопровождения и продвижения в колледже 5](#_Toc26543869)

[**Практическая часть** 6](#_Toc26543870)

[Создание локального способа компиляции программного кода на Си 6](#_Toc26543871)

[Вставка ассемблерного кода 9](#_Toc26543872)

[Дизассемблирование исполняемого файла 10](#_Toc26543873)

[Отчёт по тестированию 11](#_Toc26543874)

[Написание 2-х консольных программ на C# 11](#_Toc26543875)

[Составление руководства пользователя 12](#_Toc26543876)

[Составление отчёта по тестированию 12](#_Toc26543877)

[Общее исследование рынка IT компаний Нижнего Новгорода 16](#_Toc26543878)

[Презентация программного продукта 19](#_Toc26543879)

[**Заключение** 21](#_Toc26543880)

[Приложения: 22](#_Toc26543881)

# Введение

Учебная практика по ПМ 03 «Сопровождение и продвижение ПО отраслевой направленности» проходила в период с 25 ноября по 7 декабря на базе ГБПОУ Нижегородского Губернского колледжа.

Основной целью прохождения учебной практики является формирование у обучающихся первичных практических умений/опыта деятельности в рамках профессиональных модулей ППССЗ.

Для достижений данной цели необходимо поставить и решить следующие задачи:

* Создание локального способа компиляции программного кода на Си
* Использование ассемблерных вставок в программном коде на Си
* Дизассемблирование исполняемого файла
* Тестирование программ
* Написание 2-х консольных программ
* Создание отчёта по тестированию
* Общее исследование рынка IT компаний Нижнего Новгорода
* Презентация программного продукта

# **Теоретическая часть**

## **Объект прохождения практики**

Нижегородский Губернский колледж был основан 1 сентября 2015 года. Это современное многопрофильное образовательное учреждение, которое готовит специалистов по следующим укрупненным группам специальностей: педагогика и образование, культура и искусство, информатика и вычислительная техника, экономика, информатика и вычислительная техника, право и социальное обеспечение.

Нижегородский Губернский колледж – это современное многопрофильное образовательное учреждение, которое готовит специалистов по следующим укрупненным группам специальностей: Информатика и вычислительная техника, Экономика и управление, Юриспруденция, Образование и педагогические науки, История и археология, Музыкальное искусство.

Все компьютеры колледжа объединены в единую локальную сеть. В образовательном процессе используются терминальные сервера, обеспечивающие всех обучающих единым набором новейшего программного обеспечения и вычислительных ресурсов, необходимых для учебного процесса. Информация каждого члена домена хранится в отдельном каталоге, доступ к которому ограничен при помощи настройки групповой политики и служб Active Directory. Учетная запись каждого пользователя имеет уникальный логин и пароль для авторизации, который знает только конечный пользователь. Таким образом организована защита от посторонней авторизации.

Все компьютеры НГК имеют выход в интернет. С целью исключения возможности получить обучающимися доступ к сайтам экстремисткой направленности и иным ресурсам сети Интернет несовместимым с образовательным процессом осуществляется контентная фильтрация поступающей информации по технологии белых списков.

Исходя из всего вышесказанного, ГБПОУ НГК имеет хорошую базу для планирования и реализации проектных операций. Данная организация обладает современными информационными т техническими ресурсами.

## Выполнение сопровождения и продвижения в колледже

Сопровождение в колледже осуществляется в основном командой системных администраторов, место основной работы которых 115 кабинет. Системные администраторы выполняют сопровождение и поддержку всей системы информационных технологий колледжа: поддержка учебных терминалов, поддержка сайта колледжа и т.д.

В колледже студентами выполняется практика сопровождения через множество средств, от простых текстовых редакторов для отчётов и руководства пользователей, до специальных программ таких как “Oracle VM virtual box”, “CPU manager”, “MS visual studio” для выполнения специфичных задач.

Продвижение в колледже выполнялось Муреевой О.Н. и Куликовым Д.И., по предоставлению дополнительных образовательных услуг в виде курсов. Для они создали сайт, создали тематическую группу в социальной сети «Вконтакте», раздавали листовки и рассказывали о курсах студентам.

Практика студентов по продвижению в колледже осуществляется с помощью программ графических редакторов (Photoshop, Illustrator) и программы по презентации MS PowerPoint. Однако исследования в рамках сопровождения выполняются в программах MS Excel и MS Word.

# **Практическая часть**

## Создание локального способа компиляции программного кода на Си

Была поставлена задача найти способ локальной компиляции программного кода на Си для работы на учебных терминалах, без особенностей современных средств написания кода, таких как подсветка синтаксиса и подсказка кода. После проведения небольшого исследования было решено решить задачу через компилятор GCC (GNU Compiler Collection).

Компилятор GCC был скачен архивом “iso”, поскольку не разрешалось устанавливать эту утилиту через обычный установщик с расширением “exe”, по ссылке (http://www.equation.com/servlet/equation.cmd?fa=fortran). После скачивания была выполнена его распаковка, после чего папка программы, с содержимом указанным ниже (рис. 1), была перенесена в заготовленную папку под компилятор. Главный исполняемый файл программы находиться в папке “bin” с названием “gcc.exe”, через который потом будем выполнять компиляцию файлов.

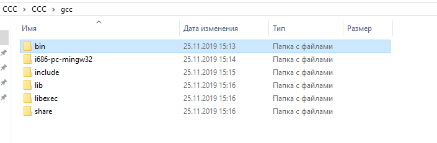


Рисунок 1 Содержимое папки GCC

Для написания кода на си был создан текстовый файл «code.txt», в котором пишется исходный код будущей программы, который нужно будет сохранять при редактировании. Для будущей демонстрации работоспособности было решено занести в него код программы вывода «hello world» (рис. 2).



Рисунок 2 текстовый файл и код примера

Компилятор GCC принимает только файлы Си кода только с соотвествующем им расширением “c”, и возвращает исполняемый файл по умолчанию в свою папку. Для выполнения этих действий надо будет выполнять много лишних действий, описанных ранее, что было бы неудобно пользователю, поэтому было принято решение по автоматизации этого процесса через скриптовый «bat» файл. Сценарий для скрипта представлен ниже (рис. 3), в начале он переходит на диск папки, далее на папку на рабочем столе, после сохраняет файл кода с расширением Си в промежуточную для компиляции папку “compl”, после передаёт этот файл компилятору и сохраняет результат компиляции в общею папку, после чего запускает его и после ставит на паузу для ознакомления с результатом компиляции программы и её выполнения.

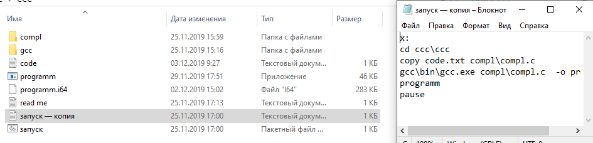


Рисунок 3 содержимое скриптового файла

Далее для удобства понимания и пользования скриптом было решено создать пользовательское руководство файлом «read me.txt» (рис. 4), в нём написан алгоритм работы с локальным компилятором.

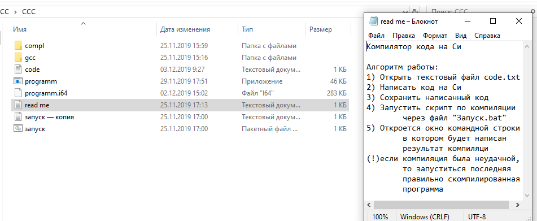


Рисунок 4 Руководство пользователя

По итогу получилась локальная утилита выполняющая поставленную задачу, снимок экрана, демонстрирующий всё содержимое предоставлен ниже (рис. 5).

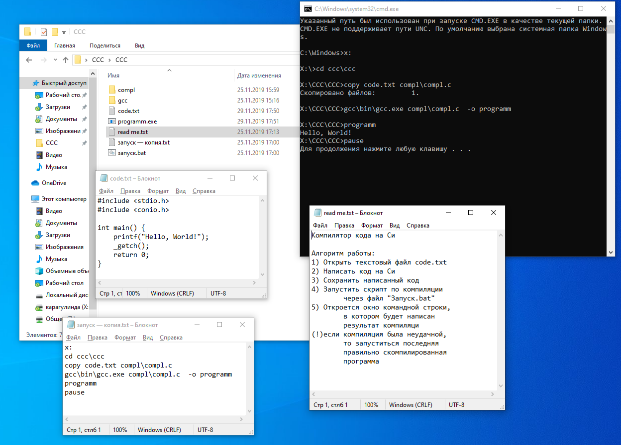


Рисунок 5 Итоговое содержимое папки компилятора

## Вставка ассемблерного кода

Вставка ассемблерного кода помогает ускорить выполнение программы за счёт доступа к низкоуровневым способам оптимизации. Вставки ассемблерного кода производились в код, написанный на Си. Вставки выполнятся с помощью команды \_\_asm, у которой так и не выявлен диалект ассемблера, которые могут быть записаны в двух форматах:

* \_\_asm инструкции ассемблера ;
* \_\_asm { assembly-instruction-list }

Примером использования ассемблерной вставки будет программа, выполняющая форматированную конкатенацию фразы «Hello world», программный код с результатами выполнения представлен ниже (рис. 6).

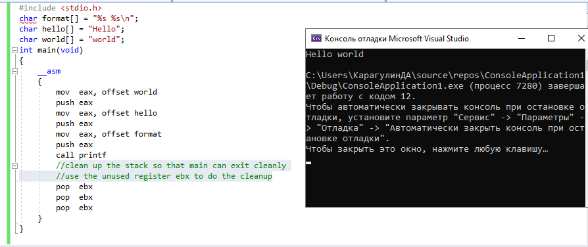


Рисунок 6 Код вставки и результат работы

Объяснение Си кода: в начале производиться подключение библиотеки, содержащей функцию вывода printf, после идёт объявление символьных массивов для сохранения нужных слов и формата, далее в main идёт вставка ассемблера.

Объяснение кода ассемблерной вставки: с начала происходит занесение значений переменных, содержащих параметры для функции printf через зарезервированный ассемблером регистр, в стек, после происходит вызов этой функции, а после очищается стек.

## Дизассемблирование исполняемого файла

Для выполнения дизассемблирования файла “exe” был выбрана ранее скомпилированная программа “hello world” на Си, с кодом которым можно ознакомиться выше (рис. 2).

Для дизассемблирования была выбрана специальная программа “IDA pro”, версии 7.0. Она показывает весь ассемблерный код программы, на рисунке ниже (рис. 7), без возможности редактирования.

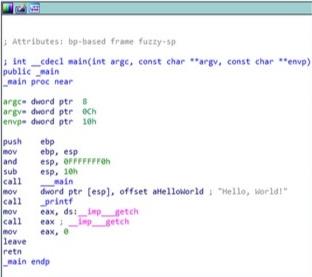


Рисунок 7 общий ассемблерный код

## Отчёт по тестированию

Отчёт по тестированию ПО должен сочетать в себе тестирования: функционирования, эргономичности, производительности по 8 программам, 4 из которых написаны одногруппником и 4 мной, 2 из которых мне ещё предстояло написать на практике вместе с руководством пользователя.

### Написание 2-х консольных программ на C#

Программы должны обладать консольным интерфейсом, а также быть понятными пользователю и должны выполнять правильно свои функции. Программы были написаны на языке программирования C#.

Программы :

1. Решение квадратных уравнений (рис. 9)
2. Калькулятор простых комбинаторных формул (рис. 8)

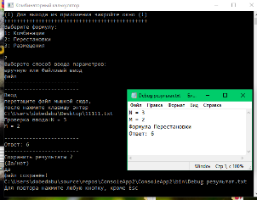


Рисунок 8 Комбинаторный калькулятор

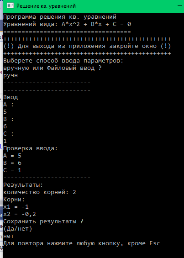


Рисунок 9 решение кв. уравнений

### Составление руководства пользователя

Руководство пользователя рассказывает, как работать с программами через графический интерфейс (рис. 10), но не рассказывает про работу с консольными версиями программ, потому что в них все инструкции прописаны.

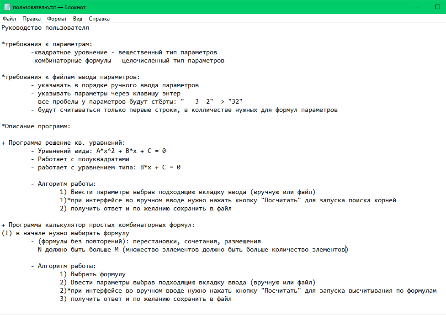


Рисунок 10 руководство пользователя

### Составление отчёта по тестированию

После написания оставшихся программ и руководства пользователя завершился предварительный этап написания отчёта, по завершению которого всё необходимое к его написанию было готово.

#### Функциональное тестирование

Функциональное тестирование можно разделить на 2 этапа: тестирование правильности ответов и тестирование функционирования программ.

Для тестирования правильности ответов были проработанный кейсы с оптимальным покрытием, составлена таблица кейсов с исходными и правильными ответами. Потом составлена таблица правильности выполнения кейсов программами. Пример таблиц ниже (рис. 11)

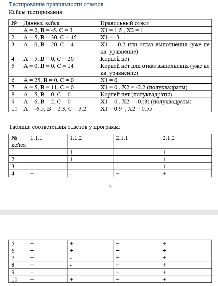


Рисунок 11 пример тестирования кейсов

Для тестирования функционирования было описаны 4 основных требования, на соответствие которым и проверялись программы (пример ниже (рис. 12)):

1. Правильность получения параметров
2. Проверка правильности данных
3. Правильность решения Задачи (выполнение всех кейсов)
4. Правильность сохранения результатов в файл



Рисунок 12 пример тестир. функционир.

#### Тестирование эргономичности

Тестирование эргономичности состоит из двух частей: описание эргономичности интерфейсов и эргономичности руководства пользователя.

Описание эргономичности интерфейсов происходило по словесному описанию пунктами в большой таблице по программам. Пример оформления для одной программы ниже (рис. 13)

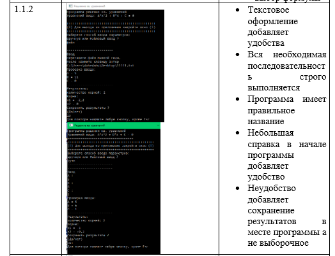


Рисунок 13пример опис. эргон. интерф.

Описание эргономичности руководства пользователя происходило комментированием руководства пользователя, в котором оценивалось понятность и покрытие всех необходимых сведений, а также замечания, если они есть, пример ниже (рис. 14)

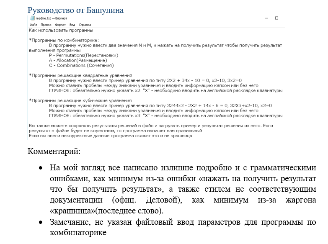


Рисунок 14 пример комментирования руков. пользователя

#### Тестирование производительности

Тестирование производительности выполнялось заполнением таблицы сведений о выполнении программы:

* время выполнения
* Потребляемая память
* Потребление ресурсов процессора

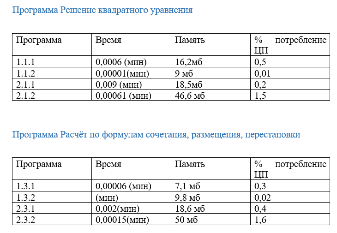


Рисунок 15 отчёт тестир. произв.

#### Заключение по отчёту

В конце отчёта выполнил общее заключение по всем тестированиям программ (рис. 16).

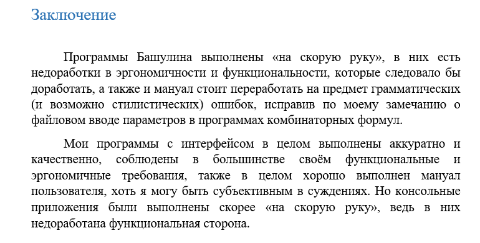


Рисунок 16 Заключение отчёта тестирования

## Общее исследование рынка IT компаний Нижнего Новгорода

Исследование проводилось через выявления среднего рейтинга IT компаний, через веб-гео-сервисы: Яндекс.Карты, Google maps, 2гис. В основном был взят ориентир на список компаний полученный данными по запросу 2Гис: «IT компании Нижнего Новгорода», не смотря на этот запрос он не смог найти таких сильных компаний в нём как: Intel, Яндекс, 1C: Первый бит. С черновым вариантом работы можно ознакомиться по ссылке в приложениях. Исследование проводилось по районам Нижнего Новгорода, с разделением на верхнею и нижнею часть города.

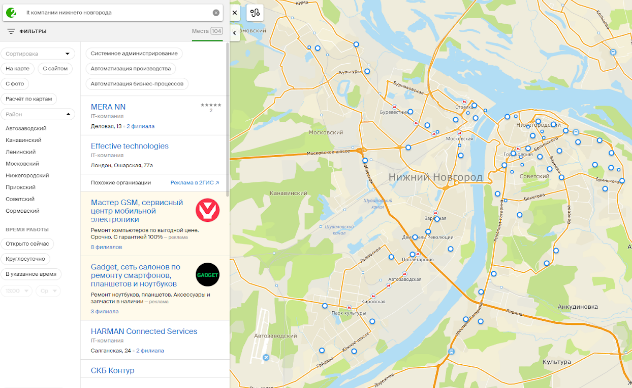


Рисунок 17 Запрос IT компаний по 2Гис

Сбор информации был поручен одногруппнику Малинкину Н.А. Всего компаний было найдено 99. Не все компании имели рейтинг, поэтому они отмечались прочерком. Пример заполнения таблицы ниже (рис. 18) или по ссылке в приложениях.



Рисунок 18 Пример таблицы сбора информации

После сбора информации по рейтингам, был проведён её анализ с помощью диаграмм, сводных таблиц, фильтров. Ниже приведён пример диаграммы по Ленинскому району (рис. 19) и сводных таблиц (рис. 20, 21).



Рисунок 19 Пример диаграммы

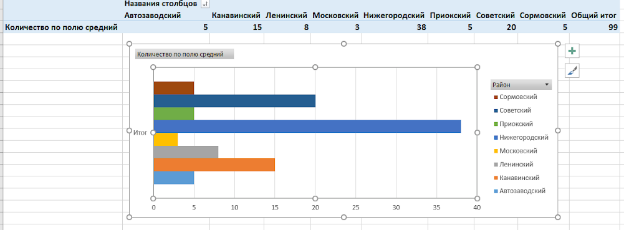


Рисунок 20 Диаграмма по сводной таблице количества компаний с рейтингом



Рисунок 21 сводная таблица по среднему рейтингу It компаний районов

Исходя из исследования было выявлено следующие:

* Наибольшее количество компаний в Нижегородском районе (38)
* Наименьшее количество компаний в Московском районе (3)
* Наибольшее количество компаний в нижней части города (61)
* Наименьшее количество компаний в верхней части города (38)
* Наибольший средний рейтинг у верхней части (4,24)
* Наименьший средний рейтинг у нижней части (~4,1)
* Общий рейтинг компаний 4,157
* Наибольший средний рейтинг у Сормовского района (4,725)
* Худший средний рейтинг у Приокского района (3,2(5))
* Расхождение среднего рейтинга между верхней и нижней частью небольшое, что говорит то, что они оба близки к общему рейтингу по городу

## Презентация программного продукта

Презентация программного продукта проводилась перед небольшой группой одногруппников и руководителем учебной практики, с помощью специальной интерактивной панели, имеющей большой экран, подходящий для презентации. Презентация создавалась и воиспроизодилась с помощью программы MS PowerPoint. Презентация происходила по сделанному ранее на учебной практике локальным компилятором СИ. Ссылка на презентацию в приложениях.

Презентация выполнена в соответствующей тематике информационных технологий, пример стиля можно рассмотреть на примере титульного слайда (рис. 22).



Рисунок 22 Титульный лист презентации

В презентации были отмечены целевые проблемы, которая она разрешает (рис. 23)

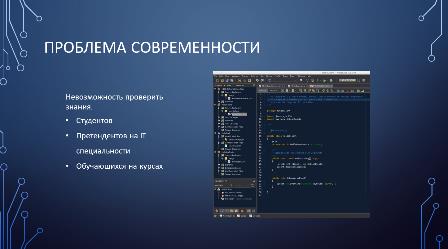


Рисунок 23 целевые проблемы

Описан программный продукт с исчерпывающими иллюстрациями, для демонстрации работы (рис. 24).

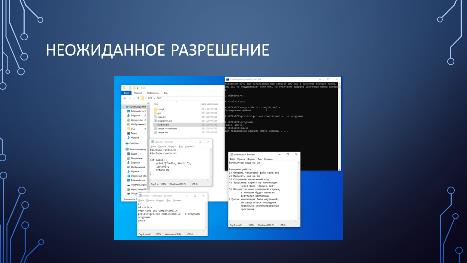


Рисунок 24 описание продукта

Также рассказано про платные дополнительные возможности программы (рис. 25).

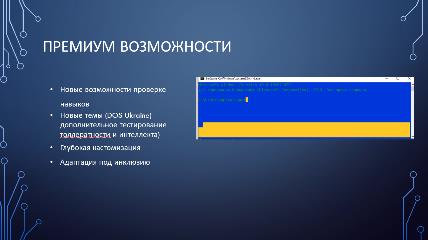


Рисунок 25 дополнительные возможности

Защита презентации программного продукта была выполнена успешно, руководителю учебной практики оценил выступление, как хорошего уровня.

# **Заключение**

В ходе прохождения учебной практики по ПМ 03 «Сопровождение и продвижение ПО отраслевой направленности» были решены все поставленные задачи.

В основном были задачи по сопровождению, имеющие техническую направленность.

В рамках продвижения было выполнено исследование рынка IT компаний Нижнего Новгорода и презентация программного продукта.

# Приложения:

* Программный код на Си для примера работы компилятора:

#include <stdio.h>

void main()

{

printf("Hello world\n");

}

* Сценарий скрипта запуска локального компилятора:

x:

cd ccc\ccc

copy code.txt compl\compl.c

gcc\bin\gcc.exe compl\compl.c -o programm

programm

pause

* Руководство пользователя от локального компилятора Си:

Компилятор кода на Си

Алгоритм работы:

1) Открыть текстовый файл code.txt

2) Написать код на Си

3) Сохранить написанный код

4) Запустить скрипт по компиляции

через файл "Запуск.bat"

5) Откроется окно командной строки,

в котором будет написан

результат компиляци

(!)если компиляция была неудачной,

то запуститься последняя

правильно скомпилированная

программа

* Код из примера ассемблерной вставки:

#include <stdio.h>

char format[] = "%s %s\n";

char hello[] = "Hello";

char world[] = "world";

int main(void)

{

\_\_asm

{

mov eax, offset world

push eax

mov eax, offset hello

push eax

mov eax, offset format

push eax

call printf

//clean up the stack so that main can exit cleanly

//use the unused register ebx to do the cleanup

pop ebx

pop ebx

pop ebx

}

}

* Ссылка на полную таблицу сбора информации:

<https://drive.google.com/file/d/115juqVtALD_8DKhiWEo4--YxM3Er7KvW/view?usp=sharing>

* Ссылка на черновой вариант отчёта по исследованию рынка IT:

<https://docs.google.com/document/d/1nq9ybVfKXtYKWyNaUlDvLrAGsn5j6W8Z/edit>

* Ссылка на презентацию программного продукта:

https://drive.google.com/file/d/1zGzI0S9B307lzN4rbUdIHeJBC3rPWgYV/view?usp=sharing