**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра МОЭВМ**

отчет

**по лабораторной работе №1**

**по дисциплине «Программирование»**

Тема: Создание Makefile и управление кодом с помощью git

|  |  |  |
| --- | --- | --- |
| Студент гр. 6304 |  | Иванов В.С. |
| Преподаватель |  | Кринкин К.В. |

Санкт-Петербург

2016

**Цель:**

Научиться создавать makefile и управлять кодом с помощью git.

**Задание:**

1. Создайте проект, состоящий из пяти файлов: main.c, print\_str.c, get\_name.c, print\_str.h, get\_name.h в каталоге, имя которого содержит Ваше имя, Фамилию и номер лабораторной.

* Файл get\_name.c должен содержать описание функции, которая считывает из входного потока имя пользователя и возвращает его.
* Файл get\_name.h должен содержать прототип функции, которая считывает из входного потока имя пользователя и возвращает его.
* Файл print\_str.c должен содержать описание функции, которая принимает в качестве аргумента строку и выводит её (функция ничего не возвращает).
* Файл print\_str.h должен содержать прототип функции, которая принимает в качестве аргумента строку и выводит её (функция ничего не возвращает).
* Файл main.c содержит главную функцию, которая вызывает функцию из файла get\_name.h, добавляет к результату выполнения функции строку “Hello,” и передает полученную строку в функцию вывода стро﻿ки из print\_str.h.

2. После того, как Ваш проект будет готов, создайте для него Makefile.

**Ход работы:**

1. **Содержимое makefile:**

*{Линковка; Исполняемый файл – main}*

main: get\_name.o print\_str.o main.o

gcc main.o -o main get\_name.o print\_str.o

*{Очистка (автоматическая)}*

rm \*.o

*{Создание объектного файла main.o}*

main.o: main.c

gcc -c main.c

*{Создание объектного файла get\_name.o}*

get\_name.o: get\_name.c get\_name.h

gcc -c get\_name.c

*{Создание объектного файла print\_str.o}*

print\_str.o: print\_str.c print\_str.h

gcc -c print\_str.c

*{Очистка (для того чтобы можно было прописать clean)}*

clean:

rm \*.o

1. **Работа с git:**
2. Git clone < https://github.com/moevm/pr1-2016-6304.git>

Так мы скопировали себе репозиторий группы 6304 для работы в нем локально

1. Git checkout –b “Ivanov\_Vladimir\_L\_1”

Это нужно для локального создания новой ветки от ветки master, чтобы загружать туда результаты лабораторных работ

1. Git push origin “Ivanov\_Vladimir\_L\_1”

Для отправки изменения на сервер, а именно в репозиторий в котором мы работали, т.е. репозиторий группы 6304.

1. Git add \*.c \*.h makefile

Таким образом, мы проиндексировали изменения.

1. Git commit –m “L\_1”

Затем создали коммит моих изменений.

1. Git push origin

Отправили изменения в репозиторий

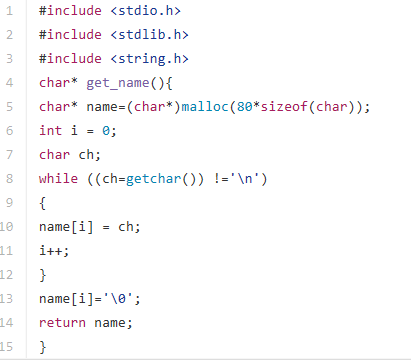
1. Создали pull request в ветку master

**Вывод:**

Выполнив данную лабораторную работу, мы поняли на практике, как написать собственный makefile для нашей работы, а также научились пользовать репозиторием GitHub.com, а именно основными командами, связанными с индексированием изменений, созданием веток и синхронизацией их с удалённым сервером.

**Приложение:**

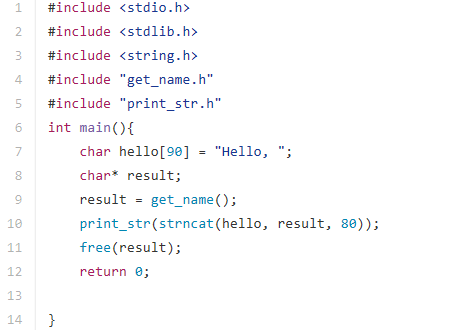
1. Get\_name.c



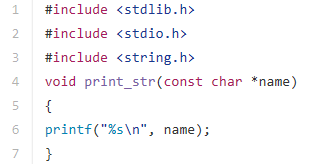
1. Get\_name.h



1. Main.c



1. Print\_str.c



1. Print\_str.h



1. Makefile

