**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра МОЭВМ**

отчет

**по лабораторной работе №1**

**Программирование**

Создание Makefile и управление кодом с помощью git

|  |  |  |
| --- | --- | --- |
| Студент гр. 6304 |  | Бочаров Ф.Д. |
| 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.

**Содержание:**

1. **Makefile:**

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

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

#Линковка

#Исполняемый файл – main

rm \*.o

#Очистка

main.o: main.c

gcc -c main.c

#Создание объектного файла main.o

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

gcc -c get\_name.c

#Создание объектного файла get\_name.o

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

gcc -c print\_str.c

#Создание объектного файла print\_str.o

clean:

rm \*.o

1. **Работа с git**
2. Git clone <https://github.com/moevm/pr1-2016-6304.git> скопируем себе репозиторий группы 6304 и будем работать в нем локально
3. Git checkout –b “Bocharov\_Filipp\_L\_1” локально создадим новую ветку от ветки master для нашего проекта
4. Git push origin “Bocharov\_Filipp\_L\_1” отправим изменения на сервер.

В данном случае за “origin” считается репозиторий группы 6304 т.к. мы работаем в нем.

1. Git add \*.c \*.h makefile проиндексируем изменения
2. Git commit –m “L\_1” создадим коммит наших изменений
3. Git push origin отправим изменения на репозиторий
4. Создадим pull request в ветку master

**Вывод:** Выполнив данную лабораторную работу, я освоил на практике написание собственного make файла для сборки проекта, а также работу с репозиторием GitHub.com, основные команды, связанные с индексированием изменений, созданием веток и синхронизации их с удалённым сервером.