**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра МОЭВМ**

отчет

**по лабораторной работе №1**

**по дисциплине «Программирование»**

Тема: Компиляция файлов при помощи make

Основы компиляции С

|  |  |  |
| --- | --- | --- |
| Студент гр. 6303 |  | Шевченко Д.В. |
| Преподаватель |  | Татьяна Т.А. |

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 // указание как правильно собирать

gcc main.o -o main get\_name.o print\_str.o // файл из имеющегося

rm \*.o // набора прототипов и функций

main.o: main.c

gcc -c main.c

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

gcc -c get\_name.c

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

gcc -c print\_str.c

clean:

rm \*.o // удаление лишних файлов созданных в процессе make

1. Getname.c

#include <stdio.h>

#include <stdlib.h>

char\* get\_name(){

char\* name = (char\*)malloc(80\*sizeof(char)); // выделение памяти

int i = 0;

char ch;

while ((ch = getchar()) != '\n')

{

name[i] = ch; // цикл считывающий данные с входного потока в массив

i++;

}

name[i] = '\0';// добавление символа конца строки

return name; // возвращение функцией указателя на строку

}

1. print\_str.c

#include <stdlib.h>

#include <stdio.h>

#include <string.h>

void print\_str(char\* name)

{

printf("%s\n", name); // функция печатает массив переданный ей

}

1. main.c

#include <string.h>

#include <stdio.h>

#include <stdlib.h>

#include "get\_name.h" // подключение библиотек и прототипов функций

#include "print\_str.h"

int main(){

char hello[90] = "Hello, "; // создание массива с текстом «Hello, »

char\* result; // создание указателя result

result = get\_name(); // указателю присваивается значение из get\_name

print\_str(strncat(hello, result, 80)); // печать значений переменных

free(result); // освобождение памяти

return 0;

}

5) get\_name.h

char\* get\_name(); // прототип функции get\_name

6) print\_str.h

char\* print\_str(const char\*); // прототип функции print\_str

Вывод: обучение созданию makefile и управлению кодом с помощью git успешно завершено.