

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)

ОТЧЕТ
по лабораторной работе № 1
по дисциплине «Программирование»
ТЕМА: СОЗДАНИЕ MAKE-ФАЙЛА

Студент гр. 7381

Адамов Я.В.

Преподаватель

Берленко Т.А.

Санкт-Петербург

2017

Лабораторная работа №1

Создание make-файла

Цель работы: научиться работать с операционной системой Linux, писать и компилировать программы на языке C, работать с системой контроля версий git.

Задание: Создать проект, состоящий из пяти файлов: main.c, print_str.c, get_name.c, print_str.h, get_name.h.

- 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.

После этого создать для проекта Makefile.

Основные теоретические положения:

Заголовочные файлы стандартной библиотеки языка C, необходимые для выполнения данной лабораторной работы: `stdio.h`, `stdlib.h`, `string.h`.

Из библиотеки `stdio.h` потребуются стандартные функции ввода и вывода, а так же функция `int puts(const char* string);`, которая выводит строку типа `char*`, на которую указывает параметр `string` в стандартный поток вывод и добавляет символ новой строки `'\n'`. Функция начинает копировать строку с адреса, указанного в `string`, пока не достигнет нулевого символа `"`. Этот заключительный, нулевой символ не копируется в стандартный поток вывод.

Из библиотеки `stdlib.h` потребуется функция `void * malloc(size_t sizemem);`, которая выделяет блок памяти, размером `sizemem` байт, и возвращает указатель на начало блока. Содержание выделенного блока памяти не инициализируется, оно остается с неопределенными значениями.

Из библиотеки `string.h` потребуется функция `char * strncat(char * destptr, char * srcptr, size_t num);`, которая добавляет первые `num` символов строки `srcptr` к концу строки `destptr`, плюс символ конца строки. Если строка `srcptr` больше чем количество копируемых символов `num`, то после скопированных символов неявно добавляется символ конца строки.

В функции `main` объявляется переменная `char hello` и указатель типа `char*` `result`, которой присваивается возвращаемое значение функции `get_name()`, которая описана в файле `get_name.c`.

В функции `get_name()` выделяется блок памяти с помощью функции `malloc`, после чего с помощью функции `getchar` вводится имя пользователь

посимвольно. Последнему элементу строки присваивается значение конца строки, которую функция возвращает.

После чего функция `main` вызывает функцию `print_str`, описанную в файле `print_str.c`, в неё подаётся функция `strncat` (которая объединяет строки), в которую в свою очередь подаются переменные `hello` и `result`. Функция `print_str` выводит результат объединения двух строк на экран.

В заголовочных файлах `get_name.h` и `print_str.h` используется `#pragma once`, что позволит избежать их повторного подключения.

Makefile содержит в себе инструкцию, по которой будет компилироваться проект.

Вывод: были освоены навыки работы с операционной системой Linux, системой контроля версий git, написания и компиляции кода на языке C, была создана программа, которая на вход принимает имя пользователя и приветствует его (выводит строку: Hello, <имя пользователя>).

Исходный код программы:

`main.c:`

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "get_name.h"
#include "print_str.h"

int main(){
    char hello[90] = "Hello, ";
    char* result;
    result = get_name();
    print_str(strncat(hello, result, 80));
    free(result);
}
```

```
        return 0;
    }
```

get_name.c:

```
#include "get_name.h"

char* get_name(){
    char* name=(char*)malloc(80*sizeof(char));
    char ch;
    int i=0;
    while((ch=getchar())!='\n'){
        name[i] = ch;
        i++;
    }
    name[i] = '\0';
    return name;
}
```

Get_name.h:

```
#pragma once

char* get_name();
```

print_str.c:

```
#include "print_str.h"

void print_str(const char *str){
    puts(str);
}
```

print_str.h:

```
#pragma once

void print_str(const char *str);
```

Makefile:

```
all: main.o get_name.o print_str.o
    gcc -o a.out main.o get_name.o print_str.o -o Yaroslav_Adamov_lr1.out
main.o: main.c get_name.h print_str.h
```

```
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
```