

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»

Студентка гр. 7381

Кревчик А.Б.

Преподаватель

Берленко Т. А.

Санкт-Петербург

2017

Цель работы.

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.

Основные теоретические положения.

1. Функция **puts**:

Прототип функции вывода строки str:

```
int puts(const char *str);
```

Заголовочный файл:

```
stdio.h
```

Описание:

Функция выводит строку в стандартный поток вывода. После вывода строки производится переход на новую строку. Символ конца строки не выводится.

2. Функция **strncat**

Прототип функции:

```
char * strncat( char * destptr, char * srcptr, num );
```

Заголовочный файл:

string.h

Описание:

Функция добавляет первые num символов строки srcptr к концу строки destptr, пока не встретится символ конца строки. Символ конца строки помещается в конце объединенных строк.

Параметры:

- **destptr**- указатель на массив в который будет добавлена строка;
- **srcptr**- строка, из которой будут копироваться первые num символов;
- **num**- максимальная длина добавляемой строки.

3. Функция **free**

Прототип функции:

```
void free( void * ptrmem );
```

Заголовочный файл:

stdlib.h

Описание

Функция free освобождает место в памяти. То есть освобожденная память может дальше использоваться программами или ОС.

Функция оставляет значение ptrmem неизменным.

Параметры:

ptrmem- указатель на ранее выделенный блок памяти, который необходимо освободить.

Возвращаемое значение:

Функция не имеет возвращаемого значение.

Вывод:

В ходе работы был изучен терминал ОС Linux, принцип работы с Make-файлом, а также функции puts, strncat, free.

Исходный код проекта:

Файл get_name.c

```
#include <stdlib.h>
#include <stdio.h>
#include "get_name.h"
char* get_name(){
char* name = (char*)malloc(80*sizeof(char));
    int i = 0;
    char ch;
    while ((ch = getchar()) != '\n')
    {
        name[i] = ch;
        i++;
    }
    name[i] = '\0';
    return name;
}
```

Файл get_name.h

```
char* get_name();
```

Файл main.c

```
#include <string.h>
#include <stdlib.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;
}
```

Файл print_str.c

```
#include <stdio.h>
#include "print_str.h"
void print_str(const char *str){
    puts(str);
}
```

Файл print_str.h

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

Файл Makefile

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