

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

Лауцюс М.

Преподаватель

Берленко Т.А.

Санкт-Петербург

2017

Цель работы.

Ознакомление с синтаксисом языка Си. Приобретение навыков создания make-файла, работы с системой контроля версий github и с терминалом операционной системы Linux.

Задание

Создание проекта, состоящего 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 создания проекта, для него должен быть создан Makefile.

Основные теоретические положения.

Заголовочные файлы стандартной библиотеки языка Си, используемые в данной лабораторной работе:

1. `stdio.h` - содержит определения макросов, константы и объявления функций и типов, используемых для различных операций стандартного ввода и вывода.

- `int puts(const char *str):`

Описание:

Функция `puts` выводит строку типа `char*`, на которую указывает параметр “string” в стандартный поток вывода и добавляет символ новой строки `'\n'`.

Функция начинает копировать строку с адреса, указанного в “string”, пока не достигнет нулевого символа `'\0'`. Этот заключительный, нулевой символ не копируется в стандартный поток вывода.

Параметры:

“string”

Си-строка для вывода на стандартный поток вывода.

Возвращаемое значение:

В случае успеха, возвращается неотрицательное значение.

В случае ошибки, функция возвращает значение EOF.

- `int getchar (FILE * filestream):`

Описание:

Получить символ из стандартного потока ввода. Функция `getchar` возвращает следующий символ из стандартного потока ввода (`stdin`).

Эта функция эквивалентна `getc`, которая принимает стандартный

поток ввода в качестве аргумента.

Параметры:

Нет

Возвращаемое значение:

Считанный символ возвращается в виде целого значения.

Если конец файла достигнут или в процессе чтения происходит ошибка, функция возвращает EOF и соответствующие индикаторы ошибки или конца файла устанавливаются.

2. string.h- заголовочный файл для работы с Си-строками.

- char * strncat(char * destptr, char * srcptr, size_t num);

Описание:

Функция 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.

Параметры:

“destptr”

Указатель на строку назначения, которая будет содержать результат конкатенации строк, включая символ завершения строки.

“srcptr”

Строка, из которой будут копироваться первые num символов для конкатенации.

“num”

Максимальное количество символов для конкатенации.

3. stdlib.h- содержит функции для преобразования чисел в текст, выделения памяти, генерации случайных чисел и др. функций-утилит.

- `void * malloc(size_t sizemem):`

Описание:

Функция `malloc` выделяет блок памяти, размером “sizemem” байт, и возвращает указатель на начало блока.

Содержание выделенного блока памяти не инициализируется, оно остается с неопределенными значениями

Параметры:

“sizemem”

Размер выделяемого блока памяти в байтах.

Возвращаемое значение:

Указатель на выделенный блок памяти. Тип данных на который ссылается указатель всегда `void*`, поэтому это тип данных может быть приведен к желаемому типу данных.

Если функции не удалось выделить требуемый блок памяти, возвращается нулевой указатель.

- `void free(void * ptrmem):`

Описание:

Функция `free` освобождает место в памяти.]

Параметры:

“ptrmem”

Указатель на блок памяти, ранее выделенный функциями `malloc`, `calloc` или `realloc`, которую необходимо высвободить. Если в качестве аргумента передается нулевой указатель, никаких действий не происходит.

Возвращаемое значение:

Функция не имеет возвращаемое значение.

Вывод:

В ходе выполнения лабораторной работы было произведено ознакомление с терминалом операционной системы Linux, с системой контроля версий github, с созданием make – файлов.

Исходный код проекта:

Файл Makefile

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

Файл get_name.c

```
#include <stdio.h>
#include <stdlib.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 print_str.c

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

Файл print_str.h

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

Файл main.c

```
#include <stdio.h>
#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;
}
```