

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

Кафедра МОЭВМ

ОТЧЕТ

по лабораторной работе №1 по дисциплине

«Программирование»

Тема: Создание make-файла

Студентка гр. 7381

Кушкеева А.О.

Преподаватель

Берленко Т.А.

Санкт-Петербург

2017

Цель работы

Ознакомиться с операционной системой Linux и системой контроля версий Git.

Изучить и понять, как работает препроцессор, компилятор и линковщик.

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`.

Основные теоретические положения

Заголовочные файлы стандартной библиотеки языка C, необходимые для выполнения данной лабораторной работы:

- `<stdio.h>` содержит прототип функции вывода строки `"int puts(const char *str);"`,

Описание:

Функция `puts` 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.

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.

- <stdlib.h> содержит прототип функции выделения и освобождения памяти “void *malloc(size_t size);”, “void free(void * ptrmem);”. Описание “void *malloc(size_t size:

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.

Описание “void free(void * ptrmem);”:

Функция free освобождает место в памяти. Блок памяти, ранее выделенный с помощью вызова malloc, calloc или realloc освобождается. То есть освобожденная память может дальше использоваться программами или ОС. Эта функция оставляет значение ptr неизменным, следовательно, он по-прежнему указывает на тот же блок памяти, а не на нулевой указатель.

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.

- <string.h> содержит прототип функции конкатенации строк “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.

Возвращаемое значение:

Указатель на строку с результатом конкатенации.

Вывод

В ходе данной лабораторной работы были изучены операционная система Linux и система контроля версий Git. Так же была освоена работа препроцессора, компилятора, линковщика.

Исходный код

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.h

```
#pragma    once
char* get_name();
```

get_name.c

```
#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;
}
```

print_str.h

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

print_str.c

```
#include <stdio.h>
#include "print_str.h"
```

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

Makefile

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