

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

ОТЧЕТ
по лабораторной работе №1
по дисциплине «Программирование»
Тема: «Создание make-файла»

Студент гр. 7381

Трушников А.П.

Преподаватель

Берленко Т.А.

Санкт-Петербург

2017

Цель работы.

Изучить основы сборки Makefile (писать цели, инструкции и команды), научиться пользоваться терминалом в ОС Linux, компилировать исходные файлы, создавать заголовочные файлы. Познакомиться с системой контроля версий github.

Основные теоретические положения.

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

1. Функция «void print_str(char* string)». В этой функции заголовочный файл «stdio.h» (стандартный заголовочный файл ввода-вывода) подключает прототип функции «int puts(const char *str)», которая склеивает строку приветствия и имя.

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:

«char* string» - Си-строка для вывода на стандартный поток вывода.

Возвращаемое значение:

В случае успеха, возвращается неотрицательное значение. В случае ошибки, функция возвращает значение EOF.

2. Функция «int main()». В этой функции заголовочный файл «string.h» (стандартной библиотеки языка Си) подключает прототип функции «char * strncat(char * destptr, char * srcptr, size_t num)», которая склеивает строку приветствия и имя.

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.

3. Функция «char* get_name()». В этой функции заголовочный файл «stdlib.h» подключает прототип функции «void * malloc(size_t sizemem)», которая склеивает строку приветствия и имя.

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.

4. Функция «int main()». В этой функции заголовочный файл «stdlib.h» подключает прототип функции «void free(void* ptrmem)».

Описание:

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.

Обратите внимание, что 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.

Выводы.

Во время работы над проектом познакомился с ОС Linux. Научился пользоваться терминалом (создавать новые директории, перемещаться между ними, использовать текстовый редактор, создавать новые файл), так же - компилировать исходные файлы (gcc), создавать объектный файл (gcc -o), заголовочный файл (*.h), «Makefile» (писать цели, зависимости, команды). Узнал, что такое система контроля версий «github». Клонировал репозиторий (git clone [url]), определял состояние файлов (git status), индексировал файлы (git add) и др.

Код Makefile:

```
all: main
```

```
main: get_name.o main.o print_str.o
```

```
    gcc get_name.o main.o print_str.o
```

```
get_name.o: get_name.c get_name.h
```

```
    gcc -c get_name.c
```

```
main.o: main.c
```

```
    gcc -c main.c
```

```
print_str.o: print_str.c print_str.h
```

```
    gcc -c print_str.c
```

