**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра** **МО ЭВМ**

отчет

**по лабораторной работе №2**

**по дисциплине «Программирование»**

Тема: Сборка программ в Си. Создание make-файла

|  |  |  |
| --- | --- | --- |
| Студентка гр. 8383 |  | Кормщикова А. О. |
| Преподаватель |  | Чайка К. В. |

Санкт-Петербург

2018

**Цель:** Разобраться в работе утилиты make, синтаксисе скрипта для Makefile. Создание рабочего make-файла для сборки программы menu, которая выполняет одну из четырех задач.

**Основные теоретические положения**

Makefile - список инструкций для утилиты make, которая позволяет собирать проект сразу целиком. Программа make с помощью этого файла позволяет автоматизировать процесс компиляции программы и выполнять при этом различные действия. При запуске make по умолчанию ищет файл Makefile в текущей папке и обрабатывает его (можно изменить это поведение, чтобы открывался другой файл с набором инструкций, если ввести команду make -f другое\_имя\_makefile).

1. Любой make-файл состоит из
2. списка целей
3. зависимостей этих целей
4. команд, которые требуется выполнить, чтобы достичь эту цель

Для сборки проекта обычно используется цель all, которая находится самой первой и является целью по умолчанию.   
  
Также, рекомендуется создание цели clean, которая используется для очистки всех результатов сборки проекта   
  
Использование нескольких целей и их зависимостей особенно полезно в больших проектах, так как при изменении одного файла не потребуется пересобирать весь проект целиком. Достаточно пересобрать измененную часть.

Объектный модуль — файл с промежуточным представлением отдельного модуля программы, полученный в результате обработки исходного кода компилятором. Объектный файл содержит в себе особым образом подготовленный код (часто называемый двоичным или бинарным), который может быть объединён с другими объектными файлами при помощи редактора связей (компоновщика) для получения готового исполнимого модуля либо библиотеки.  
В программировании заголовочный файл (англ. header file) или подключаемый файл — файл, содержимое которого автоматически добавляется препроцессором в исходный текст в том месте, где располагается некоторая директива ( [#include](https://vk.com/im?sel=62665380&st=%23include) <file.h> в Си).   
  
В языках программирования Си и C++ заголовочные файлы — основной способ подключить к программе типы данных, структуры, прототипы функций, перечислимые типы и макросы, используемые в другом модуле. По умолчанию используется расширение .h

**Реализация:**

Напишем функции для каждой отдельной подзадачи. В зависимости от команды функции должны выводить следующее:

0: индекс первого отрицательного элемента. (index\_first\_negative.c)   
1: индекс последнего отрицательного элемента. (index\_last\_negative.c)   
2 : Найти произведение элементов массива, расположенных от первого отрицательного элемента (включая элемент) и до последнего отрицательного (не включая элемент). (multi\_between\_negative.c)   
3 : Найти произведение элементов массива, расположенных до первого отрицательного элемента (не включая элемент) и после последнего отрицательного (включая элемент). (multi\_before\_and\_after\_negative.c)   
иначе необходимо вывести строку "Данные некорректны".

Напишем функции menu, в которой будут вызываться каждая из данных функций.

Создадим заголовочные файлы к функциям.

В Makefile прописываем задачи all, clear, menu.o, indexFN.o, indexLN.o, multiBAAN.o, multiBN.o, menu

**Тесты:**

1)

Sample Input:

0 -5 -3 4 3 2 5 -8 3 -9 -3

Sample Output:

0

2)

Sample Input:

0 5 6 3 1 0 8 7

Sample Output:

Данные некорректны

3)

Sample Input:

1 5 4 2 -3 2 3 -4 5 2 1

Sample Output:

6

4)

Sample Input:

2 5 3 5 -8 3 9 3 -7

Sample Output:

-648

5)

Sample Input:

3 2 1 2 -3 4 2 1 -2 2 10

Sample Output:

-160

6)

Sample Input:

3 1 -8 9 7 -5 6 1 0

Sample Output:

0

7)

Sample Input:

3

Sample Output:

Данные некорректны

**Вывод.**

Полученные знания помогли разобраться в механизме реализации сборки программы на языке СИ, с их помощью был написан make-файл, который позволяет упростить задачу повторной компиляции и позволяет собирать проект сразу целиком.

**Приложение (Листинг программы):**

**1) Makefile**

all: menu

clear:

rm menu \*.o

menu.o: menu.c

gcc -c -o menu.o menu.c

indexFN.o: index\_first\_negative.c

gcc -c -o indexFN.o index\_first\_negative.c

indexLN.o: index\_last\_negative.c

gcc -c -o indexLN.o index\_last\_negative.c

multiBAAN.o: multi\_before\_and\_after\_negative.c indexFN.o indexLN.o

gcc -c -o multiBAAN.o multi\_before\_and\_after\_negative.c

multiBN.o: multi\_between\_negative.c indexFN.o indexLN.o

gcc -c -o multiBN.o multi\_between\_negative.c

menu: menu.o indexFN.o indexLN.o multiBAAN.o multiBN.o

gcc -o menu menu.o indexFN.o indexLN.o multiBAAN.o multiBN.o