

**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(make, MakeFile).**

Студент гр. 8383

\_\_\_\_\_

Максимова А. А.

Преподаватель

\_\_\_\_\_

Чайка К. В.

Санкт-Петербург

2018

## Цель работы.

Научиться собирать программы с помощью утилиты **make**, используя **makefile**, понять, как в компьютере происходят: предпроцессинг, компиляция, линковка.

## Теория

Сборка проекта - это процесс получения **исполняемого файла** из **исходного кода**.

**Makefile** - список инструкций для утилиты **make**, которая позволяет собирать проект сразу целиком.

Если запустить утилиту **make**

то она попытается найти файл с именем **Makefile** в текущей директории и выполнить из него инструкции.

Если требуется задать какой-то конкретный **Makefile**, это можно сделать с помощью ключа **-f**  
**make -f AnyMakefile**

## Структура make-файла

Любой make-файл состоит из

списка **целей**  
**зависимостей** этих целей  
**команд**, которые требуется выполнить, чтобы достичь эту цель  
цель: зависимости  
[tab] команда

- Для сборки проекта обычно используется цель **all**, которая находится самой первой и является целью по умолчанию. (фактически, первая цель в файле 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 потребуются пересобирать весь проект целиком. Достаточно пересобрать измененную часть
- Таким образом, что ты выполнить цель **all**, требуется выполнить цель **hello**

- Для выполнения цели hello, а именно вызова gcc для объектных файлов, требуется что бы были выполнены цели соответствующие этим объектным файлам.
- Для выполнения цели для каждого объектного файла требуется скомпилировать исходный код
- Часто бывает необходимо изменить какие-то параметры сборки. Это может стать проблемой, если придется все изменять вручную. Что бы избежать этого, полезно использовать переменные. Для этого достаточно присвоить им значения до момента их использования и в месте использования обратиться к ним как \$ (VARNAME). Имена переменных принято писать в верхнем регистре.

### Тестирование:

0 -21 10 45 7 -23 -4 0 67 89 0 34 ----> 6

1 -20 10 0 23 -7 15 -14 8 27 67 34 23 10 0 -7 1 -8 ---->13

2 -1 18 45 0 -25 10 -15 -10 7 9 -1 10 13 0 7 12 -18 ----> 100

2 -1 18 45 -15 -10 7 9 -1 10 7 12 -18----> Операция невозможна, измените массив.  
Return=1

3 -21 10 0 -23 -7 -54 -9 10 13 14 7 0 7 12 -1 ---->51

3 -21 -7 -54 -9 10 13 14 12 -1---->Операция невозможна, измените массив. Return=1

4 85 0 94 85 9 3 2 -5 4 0 4 -56 ---->Данные некоректны

### Вывод:

Научилась создавать makefile, поняла смысл таких понятий как предпроцессинг, компиляция, трансляция, линковка.