**Отчет по лабораторной работе № 25 по курсу “Практикум на ЭВМ”**

Студент группы M8O-106Б-20 Почечура Артемий

Контакт e-mail: carbo59@yandex.ru

Преподаватель каф.806: Дубинин Алексей Владимирович

Входной контроль знаний с оценкой\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Отчет сдан\_\_\_\_\_\_\_ <<\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_>> 2020г., итоговая оценка\_\_\_\_\_

Подпись преподавателя\_\_\_\_\_\_\_\_\_\_\_\_\_

**1. Тема:** Автоматизация сборки программ с модульной структурой на языке Си с использованием утилиты make.

**2. Цель работы:** Научиться работать с утилитой make и составлять makefile для модульной программы.

**3. Задание (вариант №):** -

**4. Оборудование ПЭВМ студента:** ЭВМ MSI GL62M 7RDX, процессор Intel core I7 c ОП 8192 Мб

**5. Программное обеспечение ЭВМ студента:**

Операционная система семейства Unix, наименование Ubuntu, версия 20.04

Интерпретатор команд GNU Bash версия 4.4.20

Редактор текстов nano

**Утилиты операционной системы: -**

**6. Идея, метод, алгоритм:**

1) Для начала потребуется установить зависимости всех файлов;

2) Затем файлы из “.c” надо преобразовать в “.o” . Сделать это можно, написав правила для преобразования таких файлов;

3) Совершить линковку всех полученных файлов;

4) Очистить все промежуточные файлы;

В конце можно подумать, какие куски программы можно заменить на переменные, которые будут введены в первых строчках файла.

**7. Сценарий выполнения работы:**

Первый запуск make:

“компиляция всех файлов”

Изменение общего хедера и повторный запуск make:

“повторная компиляция всех файлов”

Изменения хэдэра, принадлежащего лишь одному файлу:

“компиляция файла, к которому принадлежит данный хэдэр”

Запуск make с названием функции очистки:

“очистка всех промежуточных файлов”

**8. Распечатка протокола:-**

**9. Дневник отладки: -**

**10. Замечание автора по существу работы: -**

**11. Выводы:** благодаря утилите make можно очень удобно компилировать несколько файлов одновременно. Большой плюс заключается в том, что при изменении одного файла, будут перекомпилированы лишь те, которые зависят от изменённого (и сам он, если это требуется), а компиляция остальных будет пропущена.

Подпись студента\_\_\_\_\_\_\_\_\_\_\_\_\_\_