

Отчет по лабораторной работе № 25 по курсу “Практикум Программирования”

Студент группы М80-109Б-22 Ефименко Кирилл Игоревич, № по списку 7

Контакты:
recrut5678@gmail.com
@vivichv9

Работа выполнена: «27» апреля 2023г.

Преподаватель: каф. 806 Сысоев МаксимАлексеевич

Отчет сдан « » _____ 20__ г., итоговая оценка ____

Подпись преподавателя

1. Тема:

Автоматизация сборки программ модульной структуры на языке C++ с использованием утилиты Make

2. Цель работы:

Ознакомиться и разобраться с утилитой Make, ее устройством, функциями, научиться писать Makefiles

3. Задание:

Написать Makefile и описать его, какая цель, что делает, что подставляется во флаги компилятора и т.д.

4. Оборудование:

Процессор AMD ryzen 7 5800U 8x 3.9GH с ОП 16384 Мб, НМД 512Гб. Монитор 1920x1080.

5. Программное обеспечение:

Операционная система семейства: *linux*, наименование: *ubuntu*, версия *18.10 cosmic*
интерпретатор команд: *bash* версия *4.4.19*.

Система программирования -- версия --, редактор текстов *emacs* версия *25.2.2*

Утилиты операционной системы -- Прикладные системы и программы --

Местонахождение и имена файлов программ и данных на домашнем компьютере --

6. Идея, метод, алгоритм решения задачи:

Написать Makefile, далее описать принципы его работы.

7. Сценарий выполнения работы:

- Более подробно читаю про Makefiles
- Пишу Makefile и описываю его

8. Распечатка протокола:

```
1  # Записываем в переменную CXX название компилятора
2  CXX      := g++
3  # Записываем в переменную CXX_FLAGS флаги компилятора
4  CXX_FLAGS := -std=c++17 -Wall -fsanitize=address
5
6  # Записываем в переменные названия папок
7  BIN      := bin # бинарные файлы (исполняемые файлы и файлы библиотек, созданные при компиляции)
8  SRC      := src # исходный код проекта файлы .cpp
9  INCLUDE  := include # папка с хэдрами с расширением .hpp
10 LIB      := lib # все библиотеки, используемые в проекте
11 TESTS    := tests # папка с тестовыми файлами
12
13 # список библиотек, которые связываются с бинарными файлами во время сборки
14 LIBRARIES := -lstdc++ -lm
15
16 # название исполняемого файла
17 EXECUTABLE := main
18
19 # название файла файла с тестами (на подоби main только для тестов)
20 RUN_TEST   := test
21
22 # цель build с зависимостью $(BIN)/$(EXECUTABLE), при вызове build выполнится зависимость,
23 # собираем исполняемый файл и кидаем в директорию bin
24 build: $(BIN)/$(EXECUTABLE)
25
26 # тут зависимость clean чистит все временные файлы, build собирает исполняемый файл
27 # и после печати Executing... выполняется собранный файл
28 run: clean build
29     @echo "Executing..."
30     $(BIN)/$(EXECUTABLE)
31
32 # тут происходит сборка проекта в команду для компиляции подставляются необходимые части
33 $(BIN)/$(EXECUTABLE): $(TESTS)/main.cpp | $(BIN)
34     @echo "Building..."
35     # консольная команда -I показывает компилятору путь к хэдрам, а -L на процессе линковки дает путь к библиотекам
36     $(CXX) $(CXX_FLAGS) -I $(INCLUDE) -L $(LIB) $^ -o $@ $(LIBRARIES)
37
38 # тут компилируются тесты в зависимость кладем исходники и хэдры файлов для тестирования
39 $(BIN)/$(RUN_TEST): $(wildcard $(TESTS)/*.cpp) $(wildcard $(TESTS)/*.hpp) | $(BIN)
40     @echo "Building tests..."
41     $(CXX) $(CXX_FLAGS) -I $(INCLUDE) -L $(LIB) $^ -o $@ $(LIBRARIES)
42
43 # создаем директорию с бинарниками
44 $(BIN):
45     mkdir $@
46
47 # удаляем все созданные в процессе сборки бинарники
48 clean:
49     @echo "Clearing..."
50     # консольная команда unix для удаления файлов
51     -rm -r $(BIN)
52
53 # здесь мы создаем цель _test чтобы ее можно было удобно вызывать из консоли и вся логика находится в зависимости
54 _test: $(BIN)/$(RUN_TEST)
55
56 # сначала происходит очистка всех бинарников при сборке, затем сборка тестов
57 # и запуск исполняемого файла с тестами
58 test: clean _test
59     @echo "Run tests..."
60     $(BIN)/$(RUN_TEST)
```

9. Дневник отладки:

№	Лаб. или дом.	Дата	Врем я	Событие	Действие по исправлению	Примечание
1	лаб	21.02.2023	17:20	Думал, что makefile сложная и ненужная штука	Сделал лабу и понял, что это не так	Мне грустно

10. Замечания автора:

Замечаний нет.

11. Выводы:

Прикольная штука makefile, было интересно автоматизировать сборку проекта, пришлось посмотреть несколько видосов на ютубчике, почитать несколько статей на Хабре и глянуть документацию по make'u когда разобрался с синтаксисом, он кажется не таким страшным, как казалось на первый взгляд. Потраченного времени не жалею теперь будет удобно собирать проект без ручного прописывания миллиона консольных команд. Недочёты при выполнении задания могут быть устранены следующим образом: --

Подпись студента
