Министерство образования Республики Беларусь

Учреждение образования «Белорусский государственный университет   
информатики и радиоэлектроники»

Факультет компьютерных систем и сетей

Кафедра информатики

Дисциплина «Операционные среды и системное программирование»

|  |  |
| --- | --- |
|  |  |

**ОТЧЕТ**

к лабораторной работе № 3

на тему «Основы программирования на С под Unix. Инструментарий программиста в Unix»

Выполнил             А. Д. Филипеня

Проверил                          Н. Ю. Гриценко

Минск 2024

**СОДЕРЖАНИЕ**

[1 Постановка задачи 3](#_Toc157722973)

[2 Краткие теоретические сведения 4](#_Toc157722974)

[3 Результаты выполнения лабораторной работы 5](#_Toc157722975)

[Выводы 6](#_Toc157722976)

[Список использованных источников 7](#_Toc157722977)

[Приложение А (обязательное) Листинг исходного кода 8](#_Toc157722978)

# **1 ПОСТАНОВКА ЗАДАЧИ**

Целью выполнения данной лабораторной работы является изучение среды программирования и основных инструментов: компилятора/сборщика gcc, управление обработкой проекта make и языка makefile. Также в ходе лабораторной работы необходимо написать программу, на языке программирования С, реализующую инвертирующий фильтр для символов. Кроме того, необходимо создать makefile для управления обработкой проекта, собрать и протестировать исполняемый файл.

# **2 КРАТКИЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ**

Язык программирования С является языком общего назначения, который широко используется для разработки системного программного обеспечения, прикладных программ и встраиваемых систем.[1]

GCC – это коллекция компиляторов для различных языков программирования, таких как С, С++, Ada, Fortran.[2]

Компилятор — это программное обеспечение, которое преобразует код, написанный на языке программирования, в машинный код, который может быть исполнен компьютером. Он является одним из основных инструментов разработки программного обеспечения и играет важную роль в создании приложений. Сборщик же создает исполняемый файл из одного или нескольких объектных файлов, полученных в результате компиляции.

Makefile - файл, содержащий набор инструкций для программы make. Программа make с помощью этого файла позволяет автоматизировать процесс компиляции программы и выполнять при этом различные действия.[3]

Для больших проектов использование makefile и утилиты make позволяет достаточно сократить время компиляции и сборки.

Для выполнения данной лабораторной работы были использованы следующие сведения и концепции:

1 Язык программирования С: код программы был полностью написан на языке программирования С.

2 Стандартные библиотеки языка С: для работы с функциями ввода и вывода, строками и символами были использованы стандартные библиотеки языка С.

3 Компилятор GCC: для компиляции и сборки программы был использован компилятор GCC.

4 Утилита make: для автоматизации процесса компиляции и сборки программы был создан makefile, в котором описаны все правила для компиляции и сборки программы.

5 Тестирование исполняемого файла: при помощи makefile и цели test был реализован способ тестирования исполняемого файла.

# **3 РЕЗУЛЬТАТЫ ВЫПОЛНЕНИЯ ЛАБОРАТОРНОЙ             РАБОТЫ**

В ходе выполнения лабораторной была разработана программа, реализующая инвертирующий фильтр для символов. Программа позволяет менять порядок символов в каждой строке потока, при этом порядок самих строк не изменяется. Длина строк ограничена константой. Результат работы программы представлен на рисунке 3.1.

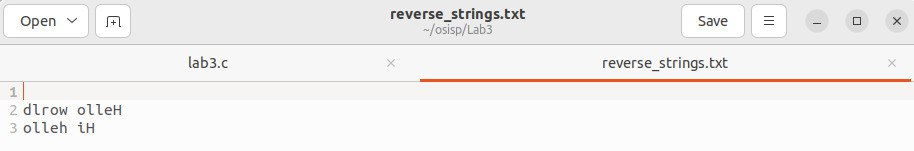


Рисунок 3.1 – Результат работы программы

При помощи утилиты make была организована возможность тестирования. Ожидаемые и фактические результаты занесены в определенные файлы перед тестированием, в процессе тестирование содержимое этих файлов сравнивается. Результат тестирования представлен на рисунке 3.2.

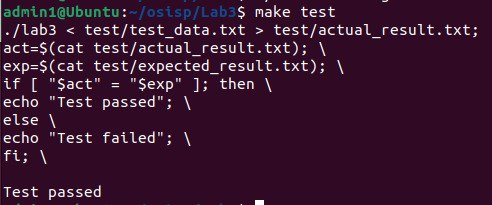


Рисунок 3.2 – Результат тестирования исполняемого файла

Таким образом, в ходе данной лабораторной работы была реализована программа, которая позволяет менять порядок символов в каждой строке потока, при этом порядок самих строк не изменяется.

# **ВЫВОДЫ**

В ходе лабораторной работы были изучена среда программирования и основные инструменты, такие как компилятор/сборщик gcc, управление обработкой проекта make и языка makefile. Кроме того, была разработана программа на языке программирования С, реализующая инвертирующий фильтр для символов. Также был создан makefile для управления обработкой проекта, а также сборки и тестирования исполняемого файла.

# **СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

[1] Язык программирования С [Электронный ресурс]. – Режим доступа: https://metanit.com/c/tutorial/1.2.php. – Дата доступа: 10.02.2024.

[2] Компилятор GCC [Электронный ресурс]. – Режим доступа: https://metanit.com/c/tutorial/1.3.php. – Дата доступа: 12.02.2024.

[3] Как устроен Makefile и что это такое? [Электронный ресурс]. – Режим доступа: https://microsin.net/programming/avr/makefile.html. – Дата доступа: 12.02.2024.

## **ПРИЛОЖЕНИЕ А**

## **(обязательное)**

## **Листинг исходного кода**

Листинг 1 – Программный код lab3.c

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#define MAX\_LINE\_LENGTH 1000

void reverseString(char \*str) {

int length = strlen(str);

for (int i = 0; i < length / 2; i++) {

char temp = str[i];

str[i] = str[length - i - 1];

str[length - i - 1] = temp;

}

}

int main() {

char buffer[MAX\_LINE\_LENGTH];

while (fgets(buffer, MAX\_LINE\_LENGTH, stdin) != NULL) {

reverseString(buffer);

printf("%s", buffer);

}

return 0;

}

Листинг 2 – Программный код makefile

CC = gcc

CFLAGS = -Wall

all: lab3

lab3: lab3.c

$(CC) $(CFLAGS) -o lab3 lab3.c

test: lab3

./lab3 < test/test\_data.txt > test/actual\_result.txt;

act=$$(cat test/actual\_result.txt); \

exp=$$(cat test/expected\_result.txt); \

if [ "$$act" = "$$exp" ]; then \

echo "Test passed"; \

else \

echo "Test failed"; \

fi; \

clean:

rm -f lab3