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

ОТЧЁТ

к лабораторной работе №3

на тему

ОСНОВЫ ПРОГРАММИРОВАНИЯ НА C ПОД UNIX. ИНСТРУМЕНТАРИЙ ПРОГРАММИСТА В UNIX.

Выполнил студент гр.153502 Толстой Д. В.

Проверил ассистент кафедры информатики Гриценко Н.Ю.

Минск 2024

**СОДЕРЖАНИЕ**

[1 Формулировка задачи 3](#_Toc158209942)

[2 Описание функций программы 4](#_Toc158209943)

[Список использованных источников 5](#_Toc158209944)

[Приложение А (обязательное) Листинг кода 6](#_Toc158209945)

1 ФОРМУЛИРОВКА ЗАДАЧИ

Целью выполнения лабораторной работы является преобразование символов потока в комбинации азбуки Морзе. Непреобразуемые символы отбрасываются.

2 ОПИСАНИЕ ФУНКЦИЙ ПРОГРАММЫ

Программа предназначена для преобразования текстового сообщения в азбуку Морзе. Программа состоит из двух основных частей: функции *text\_to\_morse*, которая выполняет преобразование текста в азбуку Морзе, и функции *main*, которая обеспечивает взаимодействие с пользователем.

Функция *text\_to\_morse* принимает на вход строку текста (*const char\* text*) и возвращает строку в азбуке Морзе, представленную в виде массива символов (*char*\*). Внутри функции определен массив *morse\_code*, содержащий коды символов в азбуке Морзе. Также определена строка *alphabet*, содержащая символы, которые могут быть преобразованы в азбуку Морзе.

Функция проходит по каждому символу входной строки. Если символ пробел, он просто добавляется в выходную строку. Если символ не пробел, то происходит поиск его индекса в строке *alphabet*, и затем копируется соответствующий ему код из массива *morse*\_*code*. В результате обработки всех символов входной строки, полученная строка в азбуке Морзе завершается символом нуля ('\0') и возвращается как результат.

Пример работы программы показан на рисунке 1.



Рисунок 1 – Пример работы программы

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

[1] Morse Code Translator [Электронный ресурс]. – Электронные данные. – Режим доступа: https://morsecode.world/international/translator.html

[2] Makefile для самых маленьких [Электронный ресурс]. – Электронные данные. – Режим доступа: https://habr.com/ru/articles/155201/

ПРИЛОЖЕНИЕ А

**(обязательное)   
Листинг кода**

main.c

#include <stdio.h>

#include <locale.h>

#include "translating.h"

int main() {

char message[512];

printf("Enter the message:\n");

fgets(message, sizeof(message), stdin);

char\* morse\_text = text\_to\_morse(message);

printf("Translated message: %s\n", morse\_text);

free(morse\_text);

return 0;

}

translating.c

#include "translating.h"

char\* text\_to\_morse(const char\* text) {

const char\* morse\_code[] = {

".-", "-...", "-.-.", "-..", ".", "..-.", "--.", "....", "..", // A-I

".---", "-.-", ".-..", "--", "-.", "---", ".--.", "--.-", ".-.", // J-R

"...", "-", "..-", "...-", ".--", "-..-", "-.--", "--..", // S-Z

"-----", ".----", "..---", "...--", "....-", ".....", "-....", "--...", "---..", "----.", // 0-9

".-.-.-", "--..--", "..--..", "-.-.--", "-....-", ".----.", "-.--.", "-.--.-", ".-...", "---...", "-.-.-.", "-...-", ".-.-.", "-....-", "..--.-", ".-..-.", ".--.-.", "-.--.-", ".-.-..", // Special characters

};

const char\* alphabet = "ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789.,?'!/()&:;=+-\_\"$@";

//const char\* alphabet = "АБВГДЕЖЗИКЛМНОПРСТУФХЦЧШЩЬЫЪЭЮЯ0123456789.,?'!/()&:;=+-\_\"$@";

char\* answer = (char\*)malloc(512 \* sizeof(char));

int j = 0;

for (int i = 0; i < strlen(text); i++) {

char c = toupper(text[i]);

if (c == ' ') {

answer[j] = ' ';

j++;

} else {

for (int k = 0; k < strlen(alphabet); k++) {

if (c == alphabet[k]) {

strcpy(answer + j, morse\_code[k]);

j += strlen(morse\_code[k]);

break;

}

}

}

}

answer[j] = '\0';

return answer;

}

makefile

CC = gcc

CFLAGS = -I.

objects = main.o translating.o

program: $(objects)

$(CC) -o program $(objects)

main.o: main.c translating.h

$(CC) $(CFLAGS) -c main.c

translating.o: translating.c translating.h

$(CC) $(CFLAGS) -c translating.c

clean:

rm -f program $(objects)

test: program

program < test\_input.txt