Министерство науки и высшего образования

Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«РЯЗАНСКИЙ ГОСУДАРСТВЕННЫЙ**

**РАДИОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ имени В.Ф. УТКИНА»**

Кафедра «Космические технологии»

Отчет о выполнении лабораторных работ

по дисциплине

«Операционные системы и системное ПО»

Выполнила:

Студентка группы 748

Брякова Г.Д.

Проверил:

доц. каф. КТ

Наумов Д.А.

Рязань 2020

Оглавление

[Лабораторная работа № 2 6](#_Toc58683292)

[Задание 1 6](#_Toc58683293)

[Задание 2 6](#_Toc58683294)

[Задание 3 6](#_Toc58683295)

[Код программы 6](#_Toc58683296)

[Задание 1 6](#_Toc58683297)

[Задание 2 7](#_Toc58683298)

[Задание 3 8](#_Toc58683299)

[Лабораторная работа № 4 9](#_Toc58683300)

[Задание 9](#_Toc58683301)

[Код программы 9](#_Toc58683302)

[Лабораторная работа № 5 11](#_Toc58683303)

[Задание 11](#_Toc58683304)

[Код программы 11](#_Toc58683305)

[Лабораторная работа № 6 13](#_Toc58683306)

[Задание 13](#_Toc58683307)

[Код программы 13](#_Toc58683308)

[Лабораторная работа № 7 14](#_Toc58683309)

[Задание 14](#_Toc58683310)

[Код программы 14](#_Toc58683311)

[Лабораторная работа № 8 15](#_Toc58683312)

[Задание 15](#_Toc58683313)

[Скриншоты о выполнении работы 16](#_Toc58683314)

Лабораторная работа № 2

Задание 1

Проверить, установлен ли первый бит числа.

Задание 2

Найти корни квадратного уравнения.

Задание 3

Вывести таблицу истинности логических функций OR, AND, XOR.

Код программы

Задание 1

#include <stdio.h>

#include <limits.h>

void main (){

int number;

printf("%s\n", "Введите число");

scanf("%i", &number);

if(number & 1){

printf("%s\n", "true");

} else {

printf("%s\n","false");

}

}

Задание 2

#include <stdio.h>

#include <math.h>

int main() {

float a, b, c;

const float EPS = 1e-5;

printf("%s\n","Write coefficient a: ");

scanf("%f", &a);

printf("%s\n","Write coefficient b: ");

scanf("%f", &b);

printf("%s\n","Write coefficient c: ");

scanf("%f", &c);

printf("%.2f \*x\*x + %.2f \* x + %.2f = 0\n", a, b, c);

printf("a = %f\n", a);

printf("b = %f\n", b);

printf("c = %f\n", c);

float d = b\*b-4\*a\*c;

printf("D = %.2f\n", d);

if (d > 0) {

float x1 = (-b+sqrtf(d))/(2\*a);

float x2 = (-b-sqrtf(d))/(2\*a);

printf("Roots are %.2f, %.2f\n", x1, x2);

} else

if (d < EPS && d >-EPS) {

float x = -b/(2\*a);

printf("Root is %.2f\n", x);

} else {

printf("No roots\n");

}

return 0;

}

Задание 3

#include <stdio.h>

int main (){

char x[]={0,0,1,1};

char y[]={0,1,0,1};

for (int i=0; i<=3; i++){

printf("%d%s%d%s%d%s", x[i], "^", y[i], " = ", x[i] && y[1], " ");

printf("%d%s%d%s%d%s", x[i], "v", y[i], " = ", x[i] || y[1], " ");

printf("%d%s%d%s%d%s\n", x[i], " XOR ", y[i], " = ", x[i] ^ y[1], " ");

}

return 0;

}

Лабораторная работа № 4

Задание

Дана непустая последовательность слов из строчных (малых) латинских букв. Слова разделены пробелом; признак конца текста – точка. Подсчитать количество слов в этом тексте.

Код программы

#include <stdio.h>

#include <string.h>

int main(){

char text[99];

printf("Write youre message: ");

gets(text);

int words = 0;

int is\_continue = 0;

char space = ' ';

for (int i = 0; text[i] != '\0'; i++){

if (text[i] != space){

is\_continue = 1;

}

if ((text[i] == space) && (is\_continue == 1)){

words++;

is\_continue = 0;

}

}

if (is\_continue == 1){

words++;

}

printf("%s%d","Count words in text: ", words);

return 0;

}

Лабораторная работа № 5

Задание

Определить, сколько раз в файле встречается заданный символ.

Код программы

#include <stdio.h>

#include <string.h>

int main(int argc, char \*\*argv) {

if(argc == 2){

if(!strcmp(argv[1], "--help")){

printf("Write the names of the files and symbol you want to count.");

return 0;

}

}

FILE \*file;

char name[10];

char symbol;

int count = 0;

printf("Write filename: ");

scanf("%s", &name);

if ((file = fopen(name, "r")) == NULL) {

printf("Can't open file\n");

printf("Press Enter");

getchar();

return 0;

} else {

printf("File %s opened succesfully\n", name);

}

printf("Write symbol, which you want to count: ");

scanf("%c\n", &symbol)

char filestring[80];

while (fgets(filestring, sizeof(filestring), file)) {

for (int i = 0; filestring[i] != '\0'; i++){

if (filestring[i] == symbol){

count++;

}

}

}

printf("Count of choosen symbol in file < %s > is equal %d.", name, count);

fclose(file);

return 0;

Лабораторная работа № 6

Задание

Создать файл, являющийся конкатенацией других файлов.

Код программы

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <sys/types.h>

#include <fcntl.h>

#include <sys/stat.h>

#define BUFFSIZE 80

int main(int argc, char const \*argv[]) {

if (argc != 4) {

printf("write your filenames for concatenate.\n");

return 0;

}

char \*buffer1 = malloc(BUFFSIZE + 1);

char \*buffer2 = malloc(BUFFSIZE + 1);

int firstF, secondF, thirdF;

if ((firstF = open(argv[1], O\_RDWR))== -1) {

perror("open failed on file");

exit(1);

} else {

printf("File %s is opened\n", argv[1]);

}

read(firstF, buffer1, BUFFSIZE);

close(firstF);

if ((secondF = open(argv[2], O\_RDWR))== -1) {

perror("open failed on file");

exit(1);

} else {

printf("File %s is opened\n", argv[2]);

}

read(firstF, buffer2, BUFFSIZE);

if ((thirdF = open(argv[3], O\_RDWR))== -1) {

perror("open failed on file");

exit(1);

} else {

printf("File %s is opened\n", argv[3]);

}

char \*buffer3 = strcat(buffer1, buffer2);

write(thirdF, buffer3, BUFFSIZE);

close(secondF);

close(thirdF);

return 0;

}

Лабораторная работа № 7

Задание

Написать и протестировать работу функции из лабораторной работы 6, создав и использовав:

* статическую библиотеку;
* динамическую библиотеку.

Создать *make*-файл(ы) для компиляции и сборки библиотек.

Код программы

MakeFile:

lb7foo.a: lb6.o

ar rv $@ $^

lb6.o: lb6.c

gcc -c $^

clean:

rm -f libfoo.a \*.o

MakeFile1:

a.out: lb6.c

gcc lb6.c

myenv: lb7foo.o libmyenv.so

gcc -o lb7foo.o -L. -lmyenv

libmyenv.so: lb7foo.o

gcc -shared -o libmyenv.so $^

lb7foo.o: lb6.c

gcc -fPIC -c $^

clean:

libmyenv.so \*.o

Лабораторная работа № 8

Задание

Изучить работу отладчика *gdb*, выполнив отладку любой собственной лабораторной работы, содержащего обращения к функциям.

* добавить в откомпилированный файл отладочную информацию;
* запустить программу под отладчиком;
* изучить команды просмотра исходного кода программы;
* изучить задание списка аргументов программе в отладчике;
* изучить команды пошагового выполнения кода;
* изучить команды вывода информации в процессе отладки;
* изучить команды работы с точками останова

Скриншоты о выполнении работы