Министерство науки и высшего образования РФ

ФГАОУ ВО Пермский национальный исследовательский

политехнический университет

Кафедра «Вычислительная математика, механика и биомеханика»

Отчет по лабораторной работе №1

Тема «Знакомство с языком программирования C»

по дисциплине «Языки программирования и методы трансляции»

Выполнил: студент группы ИСТ-22-1б Петраков М.В.

Проверил: доцент каф. ВММБ Батин С.Е.

Пермь, 2023

[Задание 1 3](#_Toc147334574)

[1.1. Постановка задания 3](#_Toc147334575)

[1.2. Решение задачи, код программы 3](#_Toc147334576)

[1.3. Тестирование работы программы 3](#_Toc147334577)

[Задание 2 4](#_Toc147334578)

[2.1. Постановка задания 4](#_Toc147334579)

[2.2. Решение задачи, код программы 4](#_Toc147334580)

[2.3. Тестирование работы программы 4](#_Toc147334581)

[Задание 3 6](#_Toc147334582)

[3.1. Постановка задания 6](#_Toc147334583)

[3.2. Решение задачи, код программы 6](#_Toc147334584)

[3.3. Тестирование работы программы 6](#_Toc147334585)

[Задание 4 8](#_Toc147334586)

[4.1. Постановка задания 8](#_Toc147334587)

[4.2. Решение задачи, код програмы 8](#_Toc147334588)

[4.3. Тестирование работы программы 8](#_Toc147334589)

[Задание 5 10](#_Toc147334590)

[5.1. Постановка задания 10](#_Toc147334591)

[5.2. Решение задачи, код программы 10](#_Toc147334592)

[5.3. Тестирование работы программы 11](#_Toc147334593)

Задание 1

1.1. Постановка задания

Написать и запустить программу, выводящую строку “Hello, World!” в терминал на языке программирования C

1.2. Решение задачи, код программы

Комментарии даются при первом использовании команды, в следующих программах идентичные команды не комментируются.

Необходимо импортировать библиотеку ввода и вывода в языке C, после чего использовать команду printf

#include <stdio.h>//загрузка стандартной библиотеки языка C

void main(){

printf("Hello, World!\n");//Вывод команды

}

1.3. Тестирование работы программы

Далее представлено использование программы. Алгоритм линейный, состоящий из одного действия.

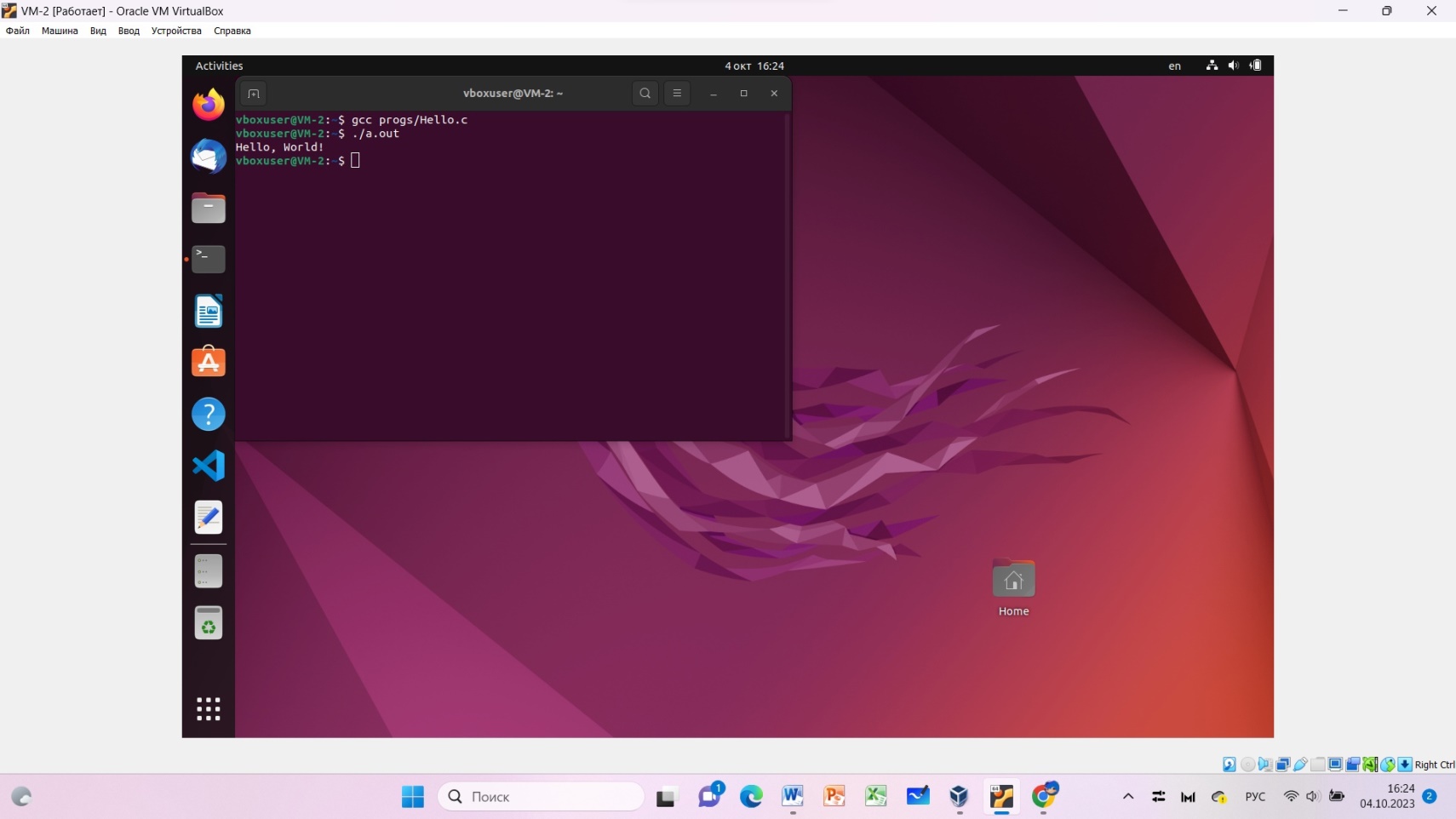


Рис.1 Успешная работа программы Hello.c

Задание 2

2.1. Постановка задания

Написать и получить результат работы программы, решающей линейное уравнение ax+b=c, коэффициенты a, b, с которого задаются при выполнении

2.2. Решение задачи, код программы

При решении уравнения ax+b=c коэффициент a не должен равняться нулю, в противном случае решения нет либо их бесконечно много, что не даёт конкретного ответа на поставленную задачу.

#include <stdio.h>

#include <float.h>

void main(){

printf("%s","Введите коэффициенты линейного уравнения ax+b=c\n");

//происходит объявление переменных, которые будут коэффициентами уравнения

double a,b,c;

//для нахождения конечного количества решений переменная a не должна равняться нулю

do{

//происходит запись переменной a до тех пор, пока пользователь не введет число, отличное от нуля

printf("%s","(a не может быть равно нулю)a=");

scanf("%lf",&a);

}while(a==0);

//вводятся значения переменных b и c

printf("%s","b=");

scanf("%lf",&b);

printf("%s","c=");

scanf("%lf",&c);

//вычисляется и выводится неизвестная x

double x=(c-b)/a;

printf("%s","x=");

printf("%lf",x);

printf("%s\n");

}

2.3. Тестирование работы программы

Далее представлен пример использования программы для решения уравнения

24x+6=18, где пользователь 3 раза пытается ввести 0 в качестве коэффициента при x.

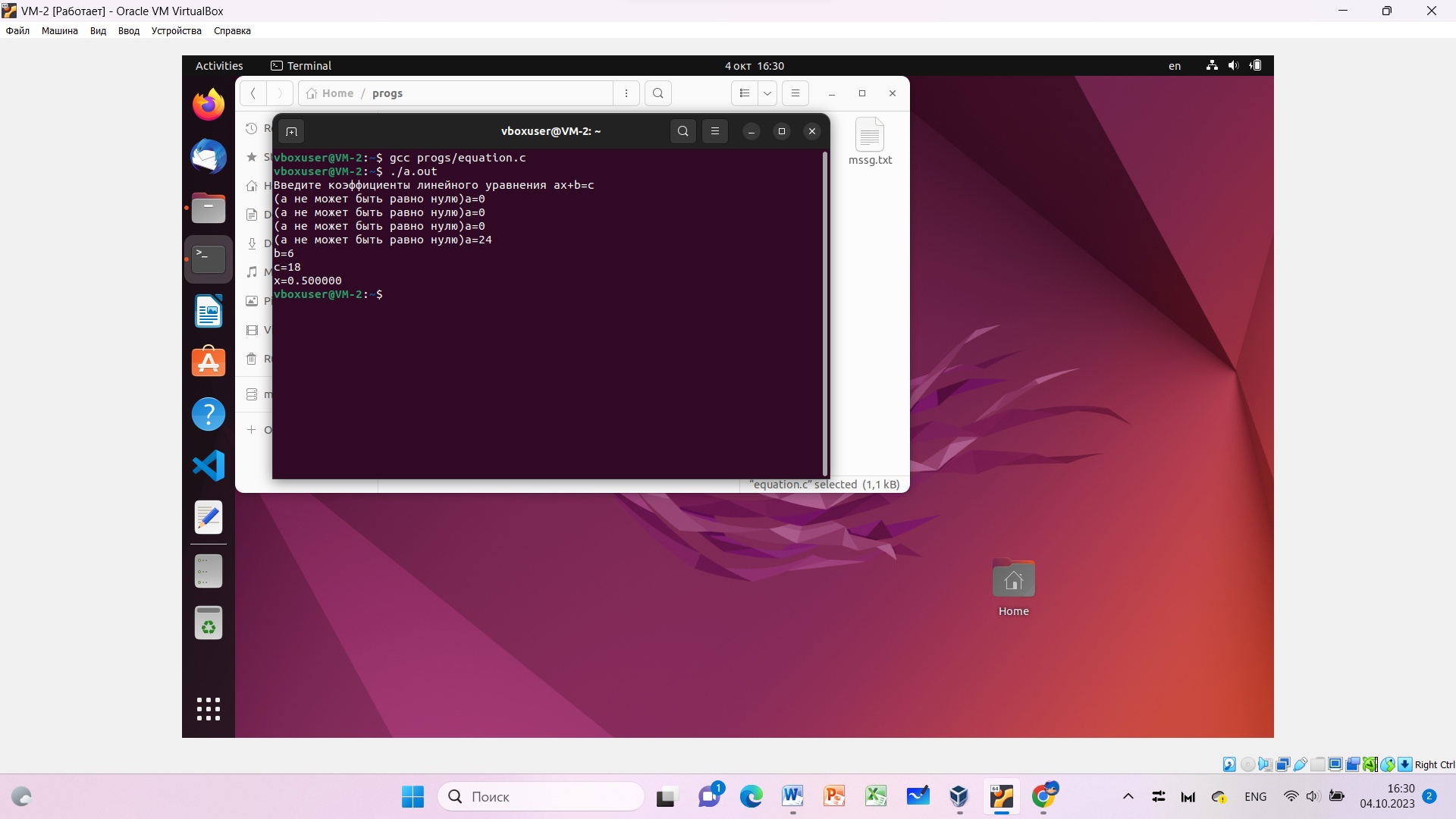


Рис.2 Успешная работа программы equation.c

Задание 3

3.1. Постановка задания

Вывести в терминал набор аргументов вызова исполняемого файла, переданных при вызове программы

3.2. Решение задачи, код программы

При вызове программы a.out считается за аргумент вызова, для хранения и вывода которых создаётся особый массив, элементы которого выводятся в цикле for. В случае введения только a.out программа уведомит об отсутствии опциональных в этом упражнении аргументов.

#include <stdio.h>

//за переменную args берется количество аргументов, введенных в командную строку

//за переменную argw - массив их наименований

int main (int args, char \* argw[]){

//создаётся переменная счётчик для цикла

int i;

for(i=0; i<args;i++){

//происходит выведение аргумента с номером i, аргумент с номером 0 - имя продукта

printf("Argument %d: %s\n", i, argw[i]);

}

if(args==1){

//в случае, если после вызова файла не были введенны аргументы, выводится сообщение об отсутствии дополнительных аргументов

printf("Коммандная строка не содержит дополнительных аргументов\n");

}

return 0;

}

3.3. Тестирование работы программы

Далее представлены примеры работы программы: в первом случае вместе с a.out вводятся дополнительные аргументы, во втором таковых нет.

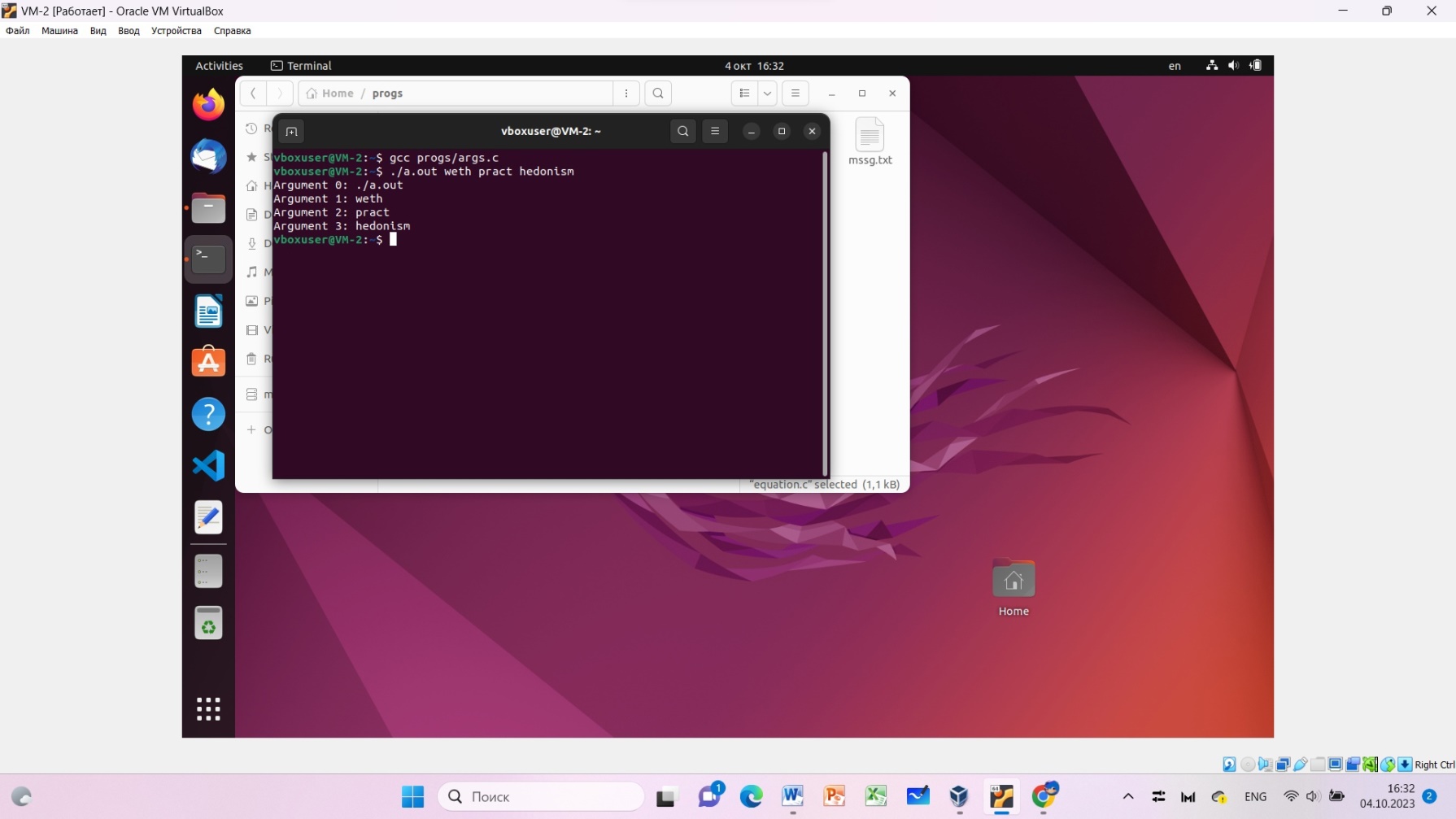


Рис.3 Работа программы args.c при введении дополнительных аргументов

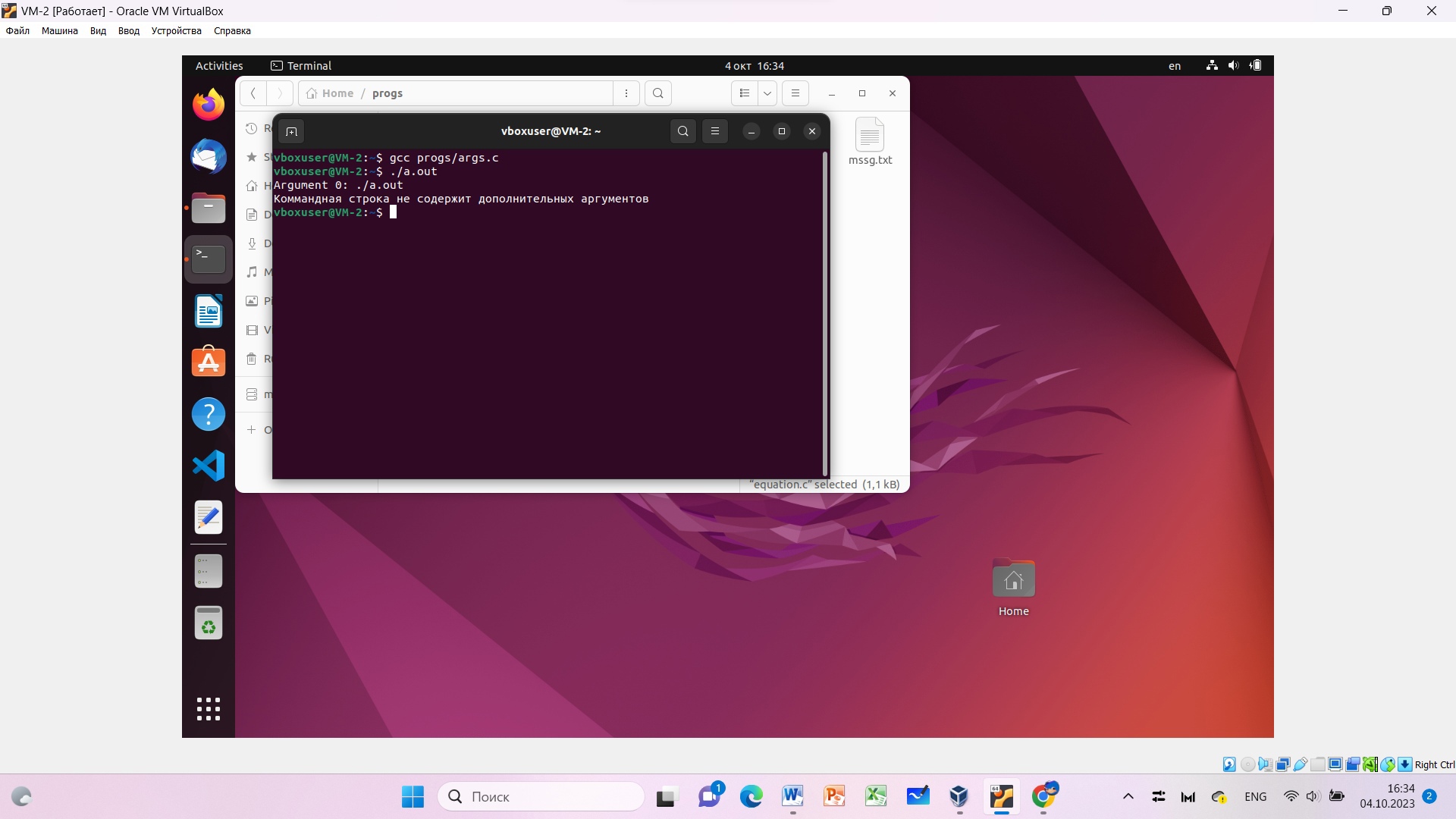


Рис. 4 Работа программы args.c при введении только a.out

Задание 4

4.1. Постановка задания

Вывести в терминал содержимое текстового файла, название которого передается при вызове исполняемого файла.

4.2. Решение задачи, код програмы

Текст будет выводится посимвольно, для этого создаётся переменная типа char, которая будет в цикле принимать значение из файла и выводить его.

#include <stdio.h>

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

//создаётся символ, через который будет выводиться текст файла

char ch;

//создаётся указатель на файл для чтения символов

FILE \*printer = fopen(name[1],"r");

if(printer==NULL){

//в случае пустого потока выводится ошибка ненахождения файла

printf("Файл не найден, перезапустите программу\n");

return 0;

}

while(ch != EOF){

//в случае успешного открытия в цикле происходит считывание символов и их вывод в терминал

ch=fgetc(printer);

printf("%c",ch);

}

//закрывается текстовый поток

fclose(printer);

return 0;

}

4.3. Тестирование работы программы

Далее приводится пример работы программы с текстовым файлом text.txt

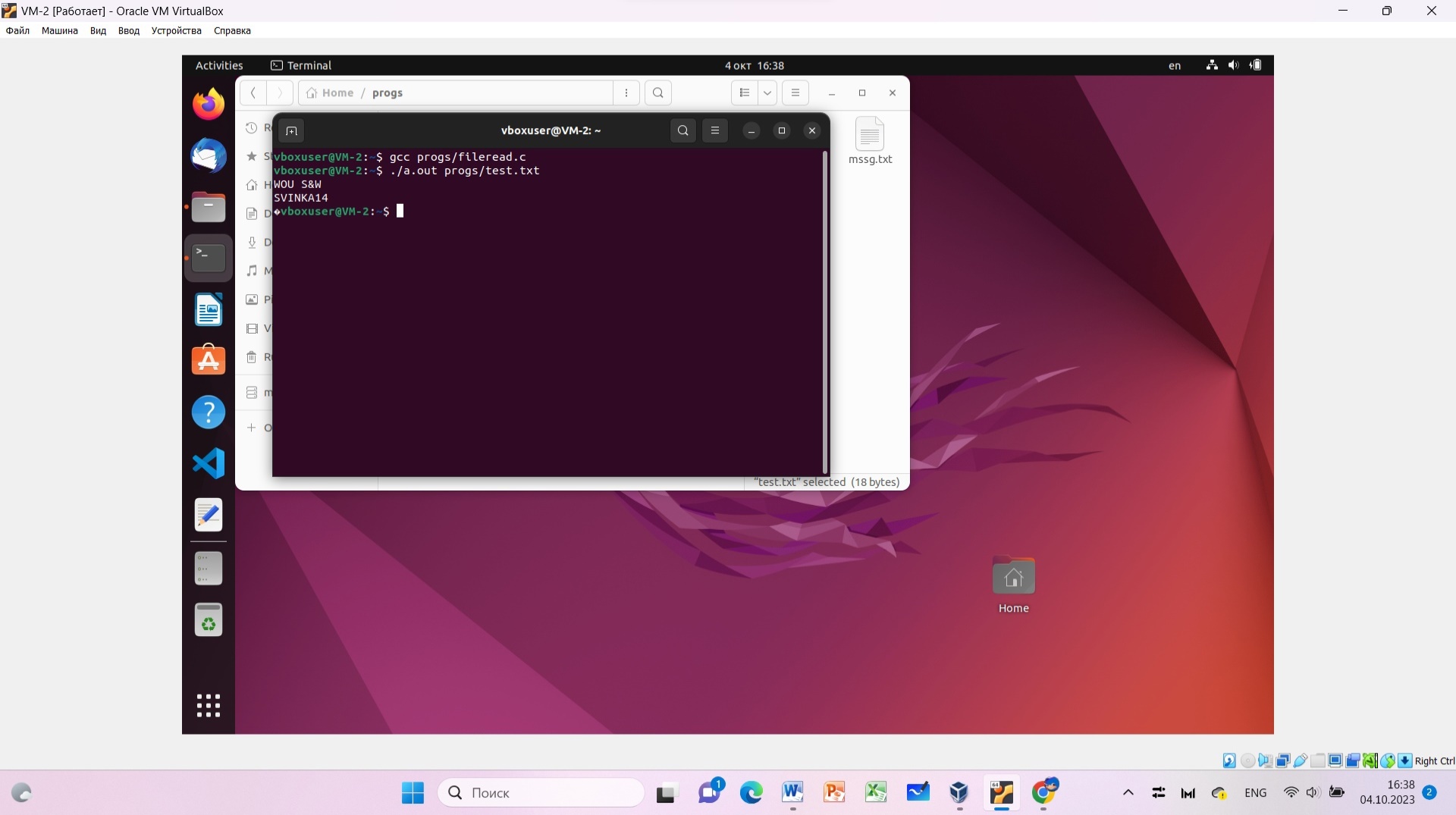


Рис.5 Пример работы программы fileread.c, чтение файла test.txt

Задание 5

5.1. Постановка задания

Вывести в начало текстового файла запрошенную при работе программы информацию и после дописать содержимое файла входных данных.

5.2. Решение задачи, код программы

В процессе работы на текстовый файл создаётся три указателя, один – для подсчёта символов, второй – для их записи, третий для редактирования. При использовании программы и работе с текстовым файлом запрашивается количество вводимых символов, а также сами символы, заносимые в будущем в начало текстового файла.

#include <stdio.h>

int main(int j,char \*name[]){

//создаётся символ для работы с изначальным текстом файла

char ch;

//создаётся переменная счётчик

int i=0;

//создаются указатели на файл для подсчёта символов, их записи, а также для записи желаемого текста

FILE \*counter, \*reader, \*writer;

//открывается поток файла для чтения, которое нужно для подсчёта количества символов

counter = fopen(name[1],"r");

if(counter==NULL){

printf("Файл не был найден, перезапустите программу\n");

return 0;

}else{

//происходит подсчёт количества символов в файле

while((ch=fgetc(counter))!= EOF){

i++;

}

fclose(counter);

}

//создаётся массив символов изначального файла

char cmass[i];

//открывается поток файла для чтения символов в массив

reader = fopen(name[1],"r");

if(reader==NULL){

printf("Файл не был найден, перезапустите программу\n");

}else{

//в цикле происходит занесение в массив символов через ранее введенную переменную

for(i=0;(ch=fgetc(reader))!=EOF;i++){

cmass[i]=ch;

}

fclose(reader);

}

//открывается поток файла для записи введенной строки и изначального набора символов файла

writer = fopen(name[1],"w");

if(writer==NULL){

printf("Файл не был найден, перезапустите программу\n");

return 0;

}else{

//создаётся переменная количества символов записываемых данных

int length;

printf("Введите количество символов записываемого предложения\n");

scanf("%d",&length);

//создаётся массив записываемых данных

char sentence[length];

printf("Введите предложение\n");

scanf("%s", sentence);

//в массив изначальных данных записывается введённый пользователем набор символов

fprintf(writer,"%s",sentence);

fprintf(writer,"\n");

//занесение в текст изначального набора символов

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

fputc(cmass[j],writer);

}

fclose(writer);

}

return 0;

}

5.3. Тестирование работы программы

Далее представлен пример работы программы. Редактируемый текстовый файл – mssg.txt, вносимая информация – “continue me”

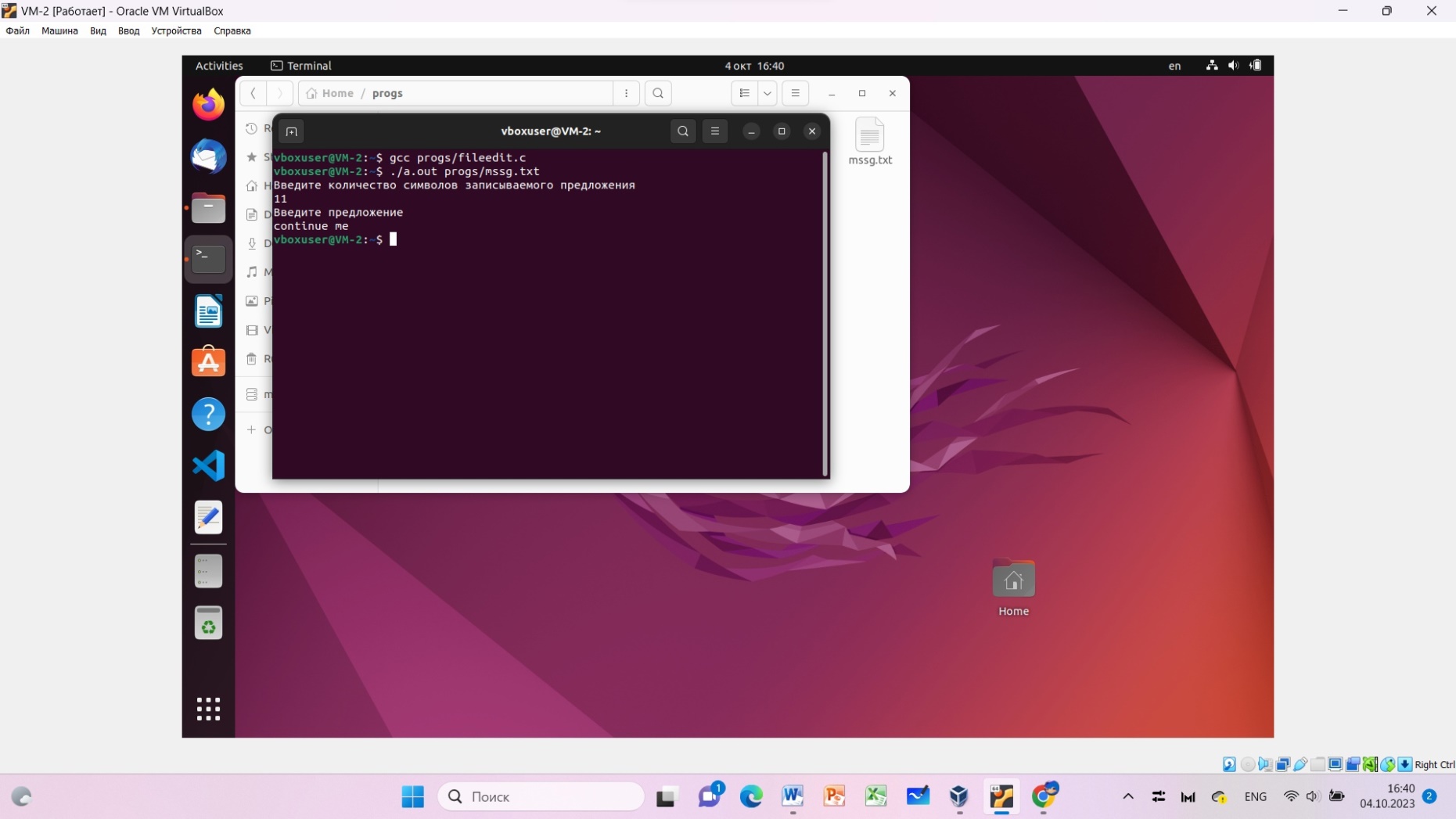


Рис.6 Пример работы программы fileedit.c с текстовым файлом mssg.txt