**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ**

**КЕМЕРОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

**Математический факультет**

**Кафедра ЮНЕСКО по новым информационным технологиям**

**ОТЧЕТ**

**О ВЫПОЛНЕНИИ ЛАБОРАТОРНОЙ РАБОТЫ №1**

«**Повтор**»

Студента

**Кузьмина Кирилла Алексеевича**

**2** курса, группы **МОА-195**

Специальность 010503 – «Математическое обеспечение

и администрирование информационных систем»

Руководитель:

(степень, звание)

И.О. Фамилия

**Антон Игоревич Зимин**

Работа защищена

«\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_»

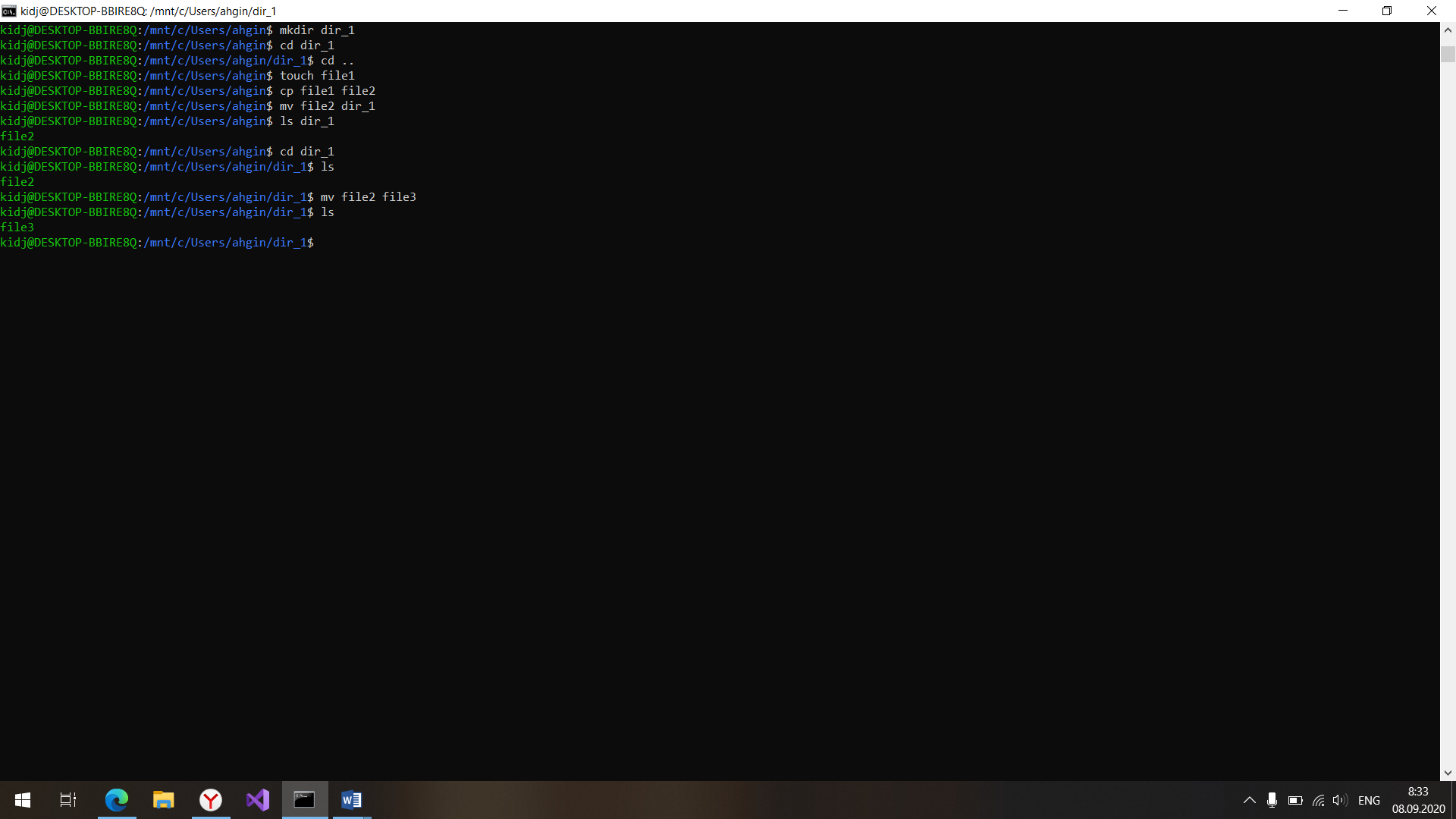
“\_\_\_\_” \_\_\_\_\_\_\_\_\_\_\_\_\_20\_\_г.

Кемерово 2019

1. Создать файл file1. Создать директорию dir1. Скопировать file1 в file2. Перенести file2 в dir1. Переименовать file2 в file3

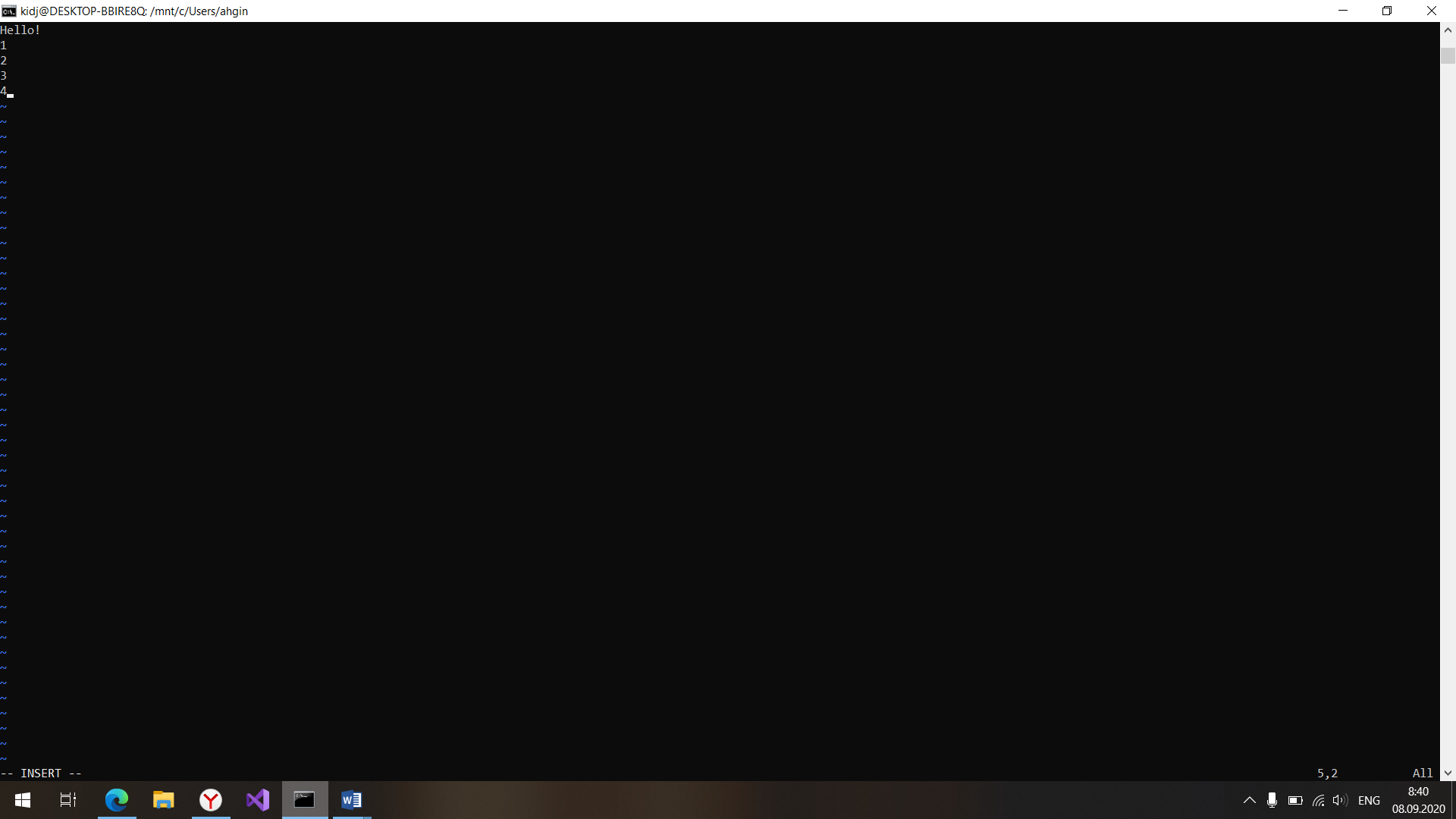
**touch file1  
mkdir dir\_1  
cp file1 file2  
mv file2 dir\_1**

**cd dir\_1  
mv file2 file3**



1. Записать текст в file1 при помощи редактора vi.

**vi file1**



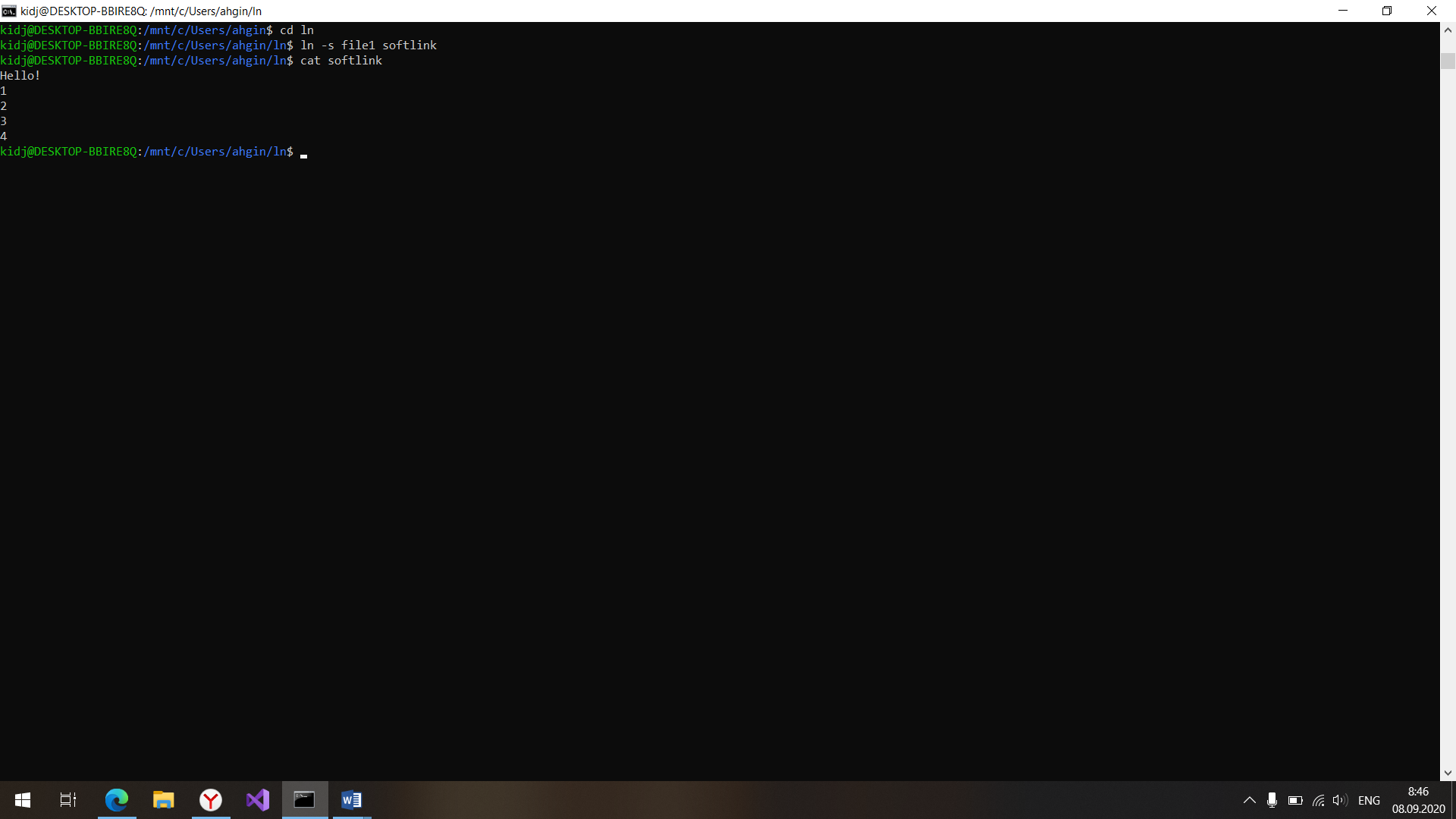
1. Создать символическую ссылку на файл file1. В чем отличие символической ссылки от жесткой?

**ln –s file1 softlink**

**Несколько отличий:**

* **символические ссылки могут ссылаться на каталоги, жёсткие нет.**
* **жёсткие ссылки имеют тот же номер inode и набор разрешений, что и сам исходный файл. Номер inode и права доступа у символических отличаются от исходного файла.**

**Главное отличие символических ссылок от жёстких в том, что при удалении файла, на который ссылаются, символическая ссылка останется, но она будет указывать в никуда, поскольку файла на самом деле больше нет.**



1. Создать файл file1. Поменять права доступа к нему на следующие:

Владелец: чтение, запись, -

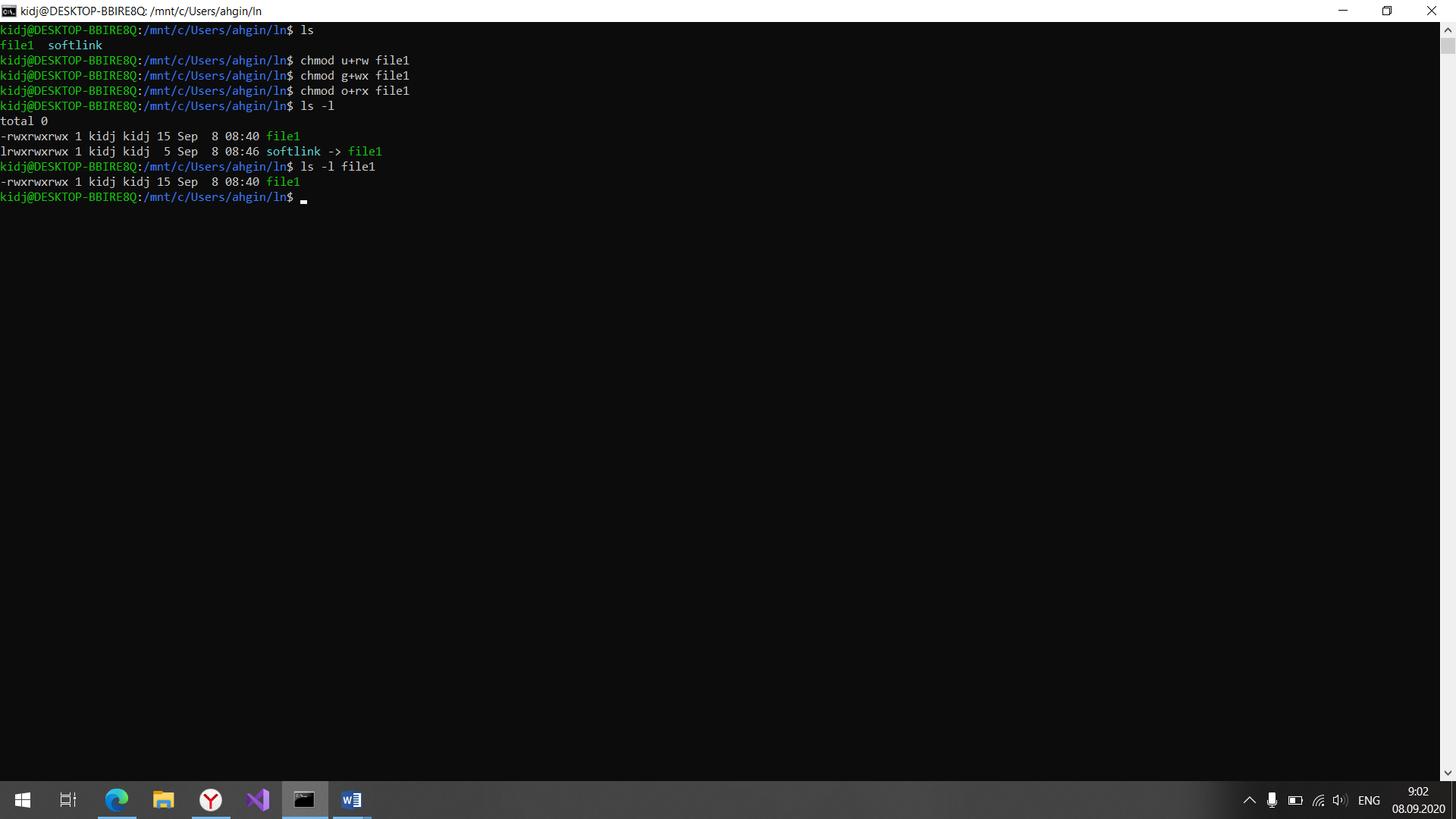
Группа: -, запись, исполнение

Остальные: чтение, -, исполнение

**chmod u+rw file1**

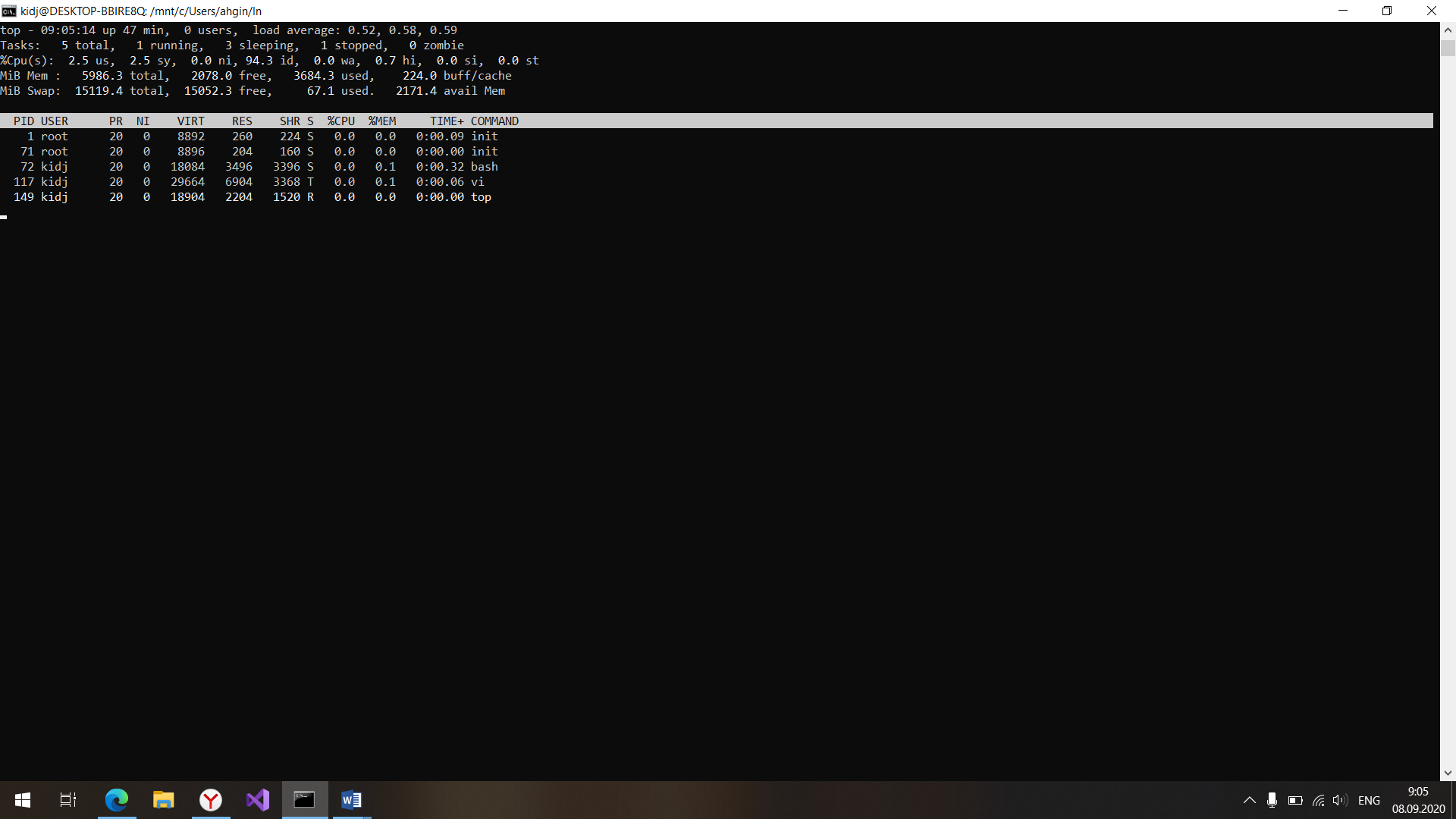
**chmod g+wx file1**

**chmod o+rx file1**



1. Вывести постоянно обновляемую информацию о всех процессах, исполняющихся в системе

**top**



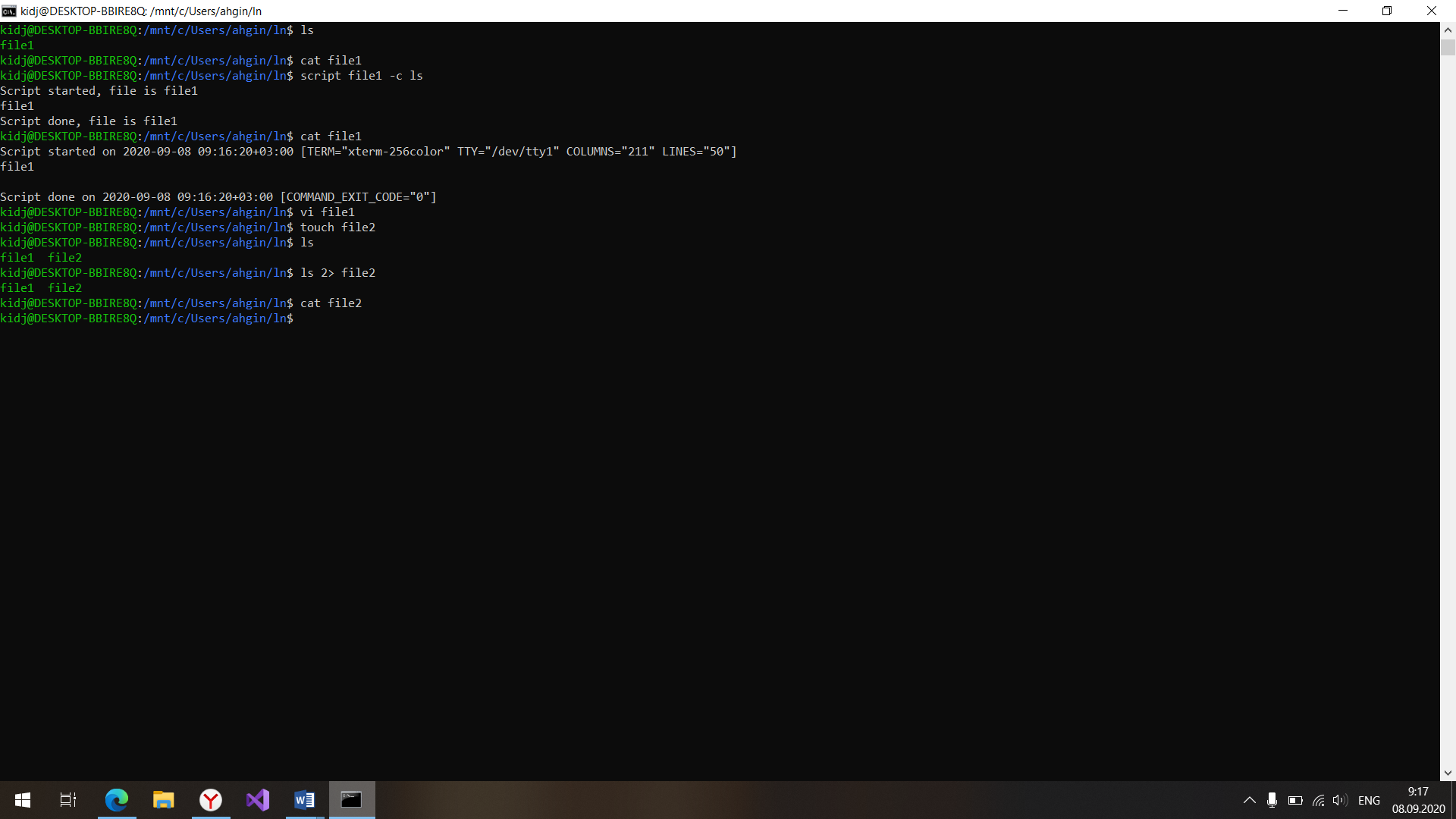
1. Записать результат выполнения команды ls в file1, а возникающие ошибки - в file2

**touch file1**

**touch file2**

**script file1 –c ls**

**ls 2> file2**



1. Написать сценарий, который выводит на экран переданные ему при запуске пользователем параметры.

**#!/bin/bash**

**count=1**

**n=$#**

**for ((i = 0; i < $n; i++))**

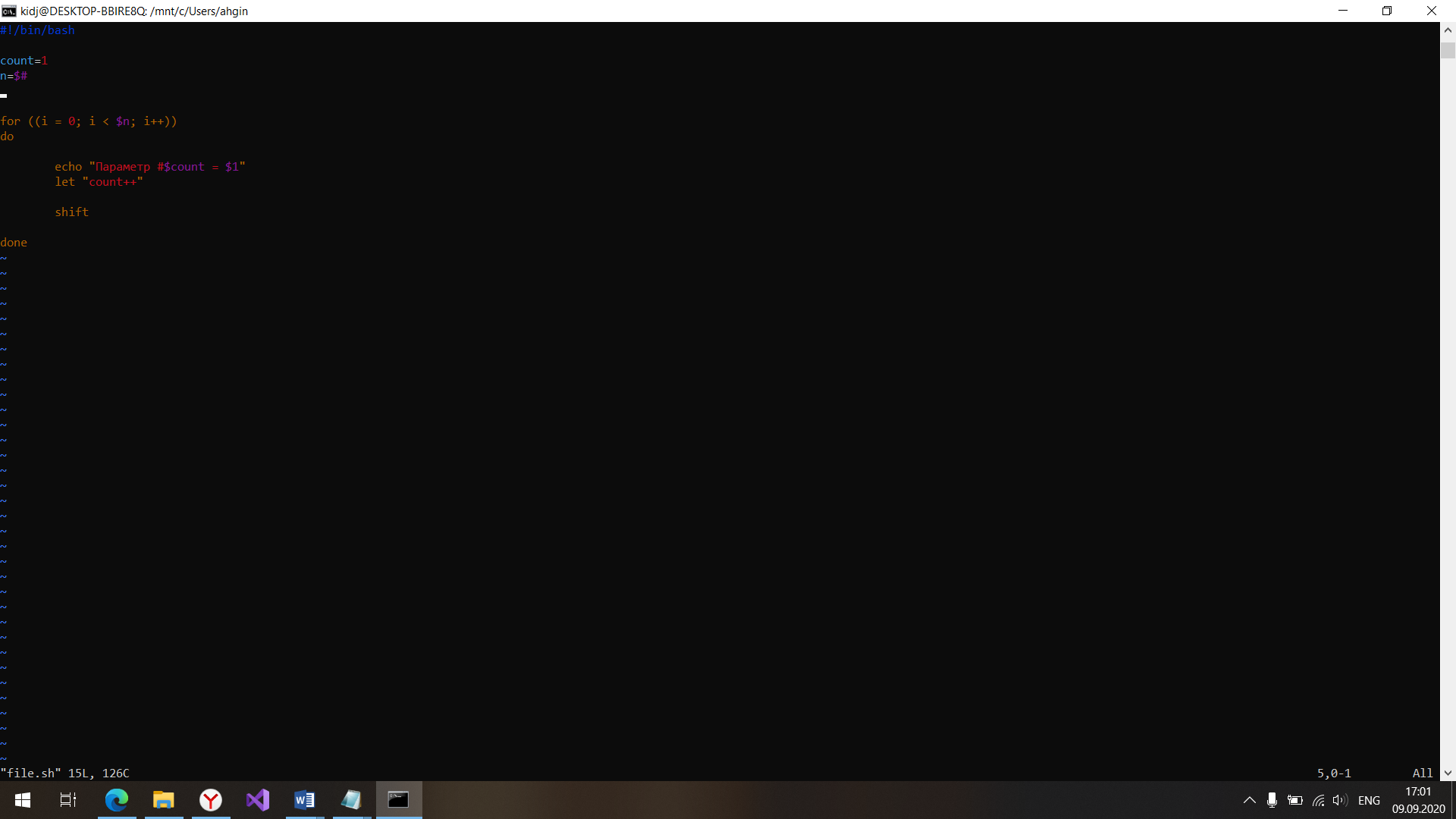
**do**

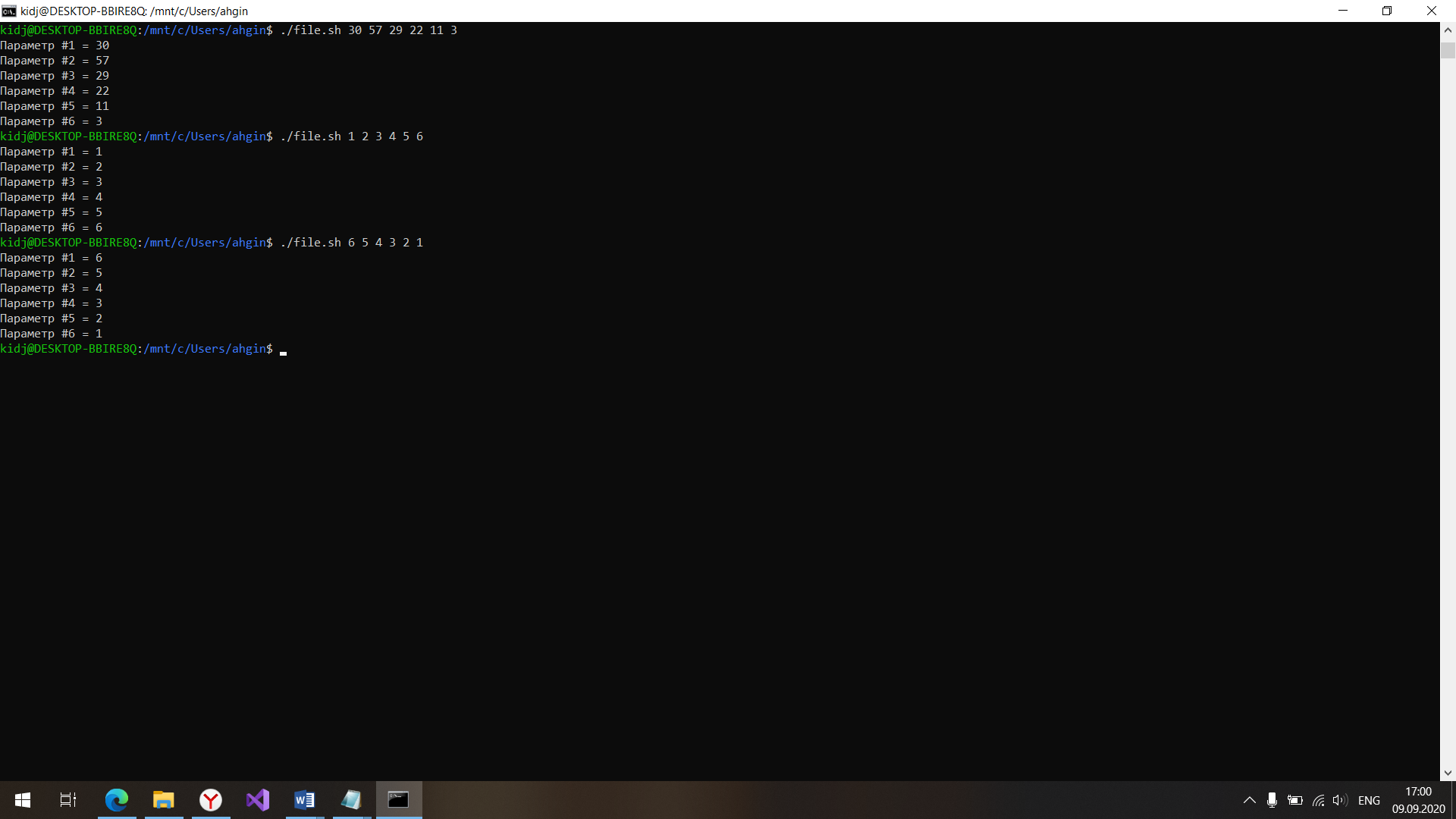
**echo "Параметр #$count = $1"**

**let "count++"**

**shift**

**done**

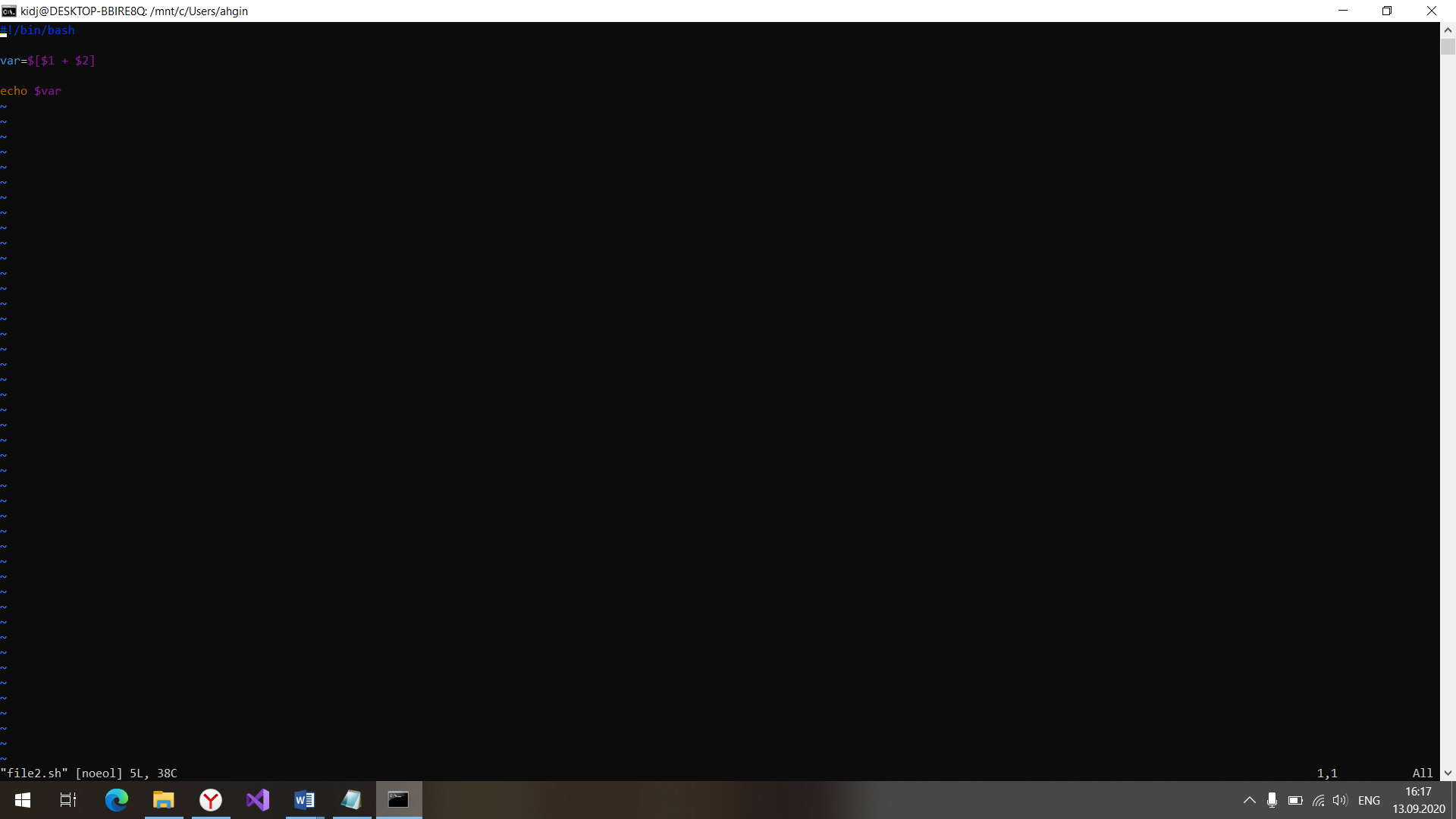


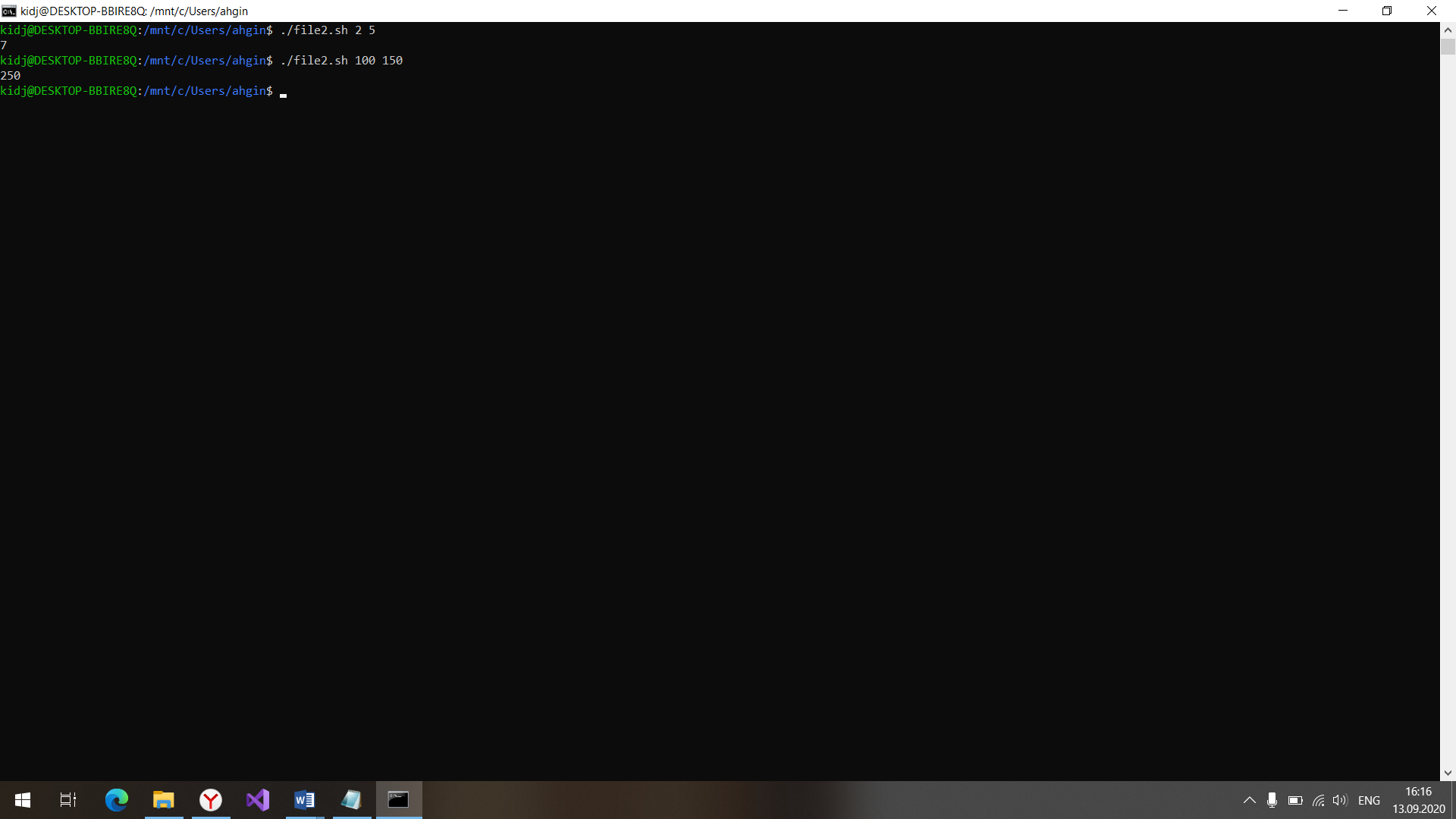


1. Написать сценарий, которому в качестве параметров передаются два числа, а он выводит на экран их сумму.

**#!/bin/bash**

**var=$[$1 + $2]**

**echo $var**



1. Написать сценарий, которому в качестве параметров передаются неограниченное количество чисел, а он выводит на экран их сумму. Пример запуска:

./scen.sh 10 2 8 90 56

Для организации перебора понадобится команда shift и переменная $#

**#!/bin/bash**

**sum=0**

**count=$#**

**for ((i = 0; i < $count; i++))**

**do**

**arr[$i]=$1**

**shift**

**done**

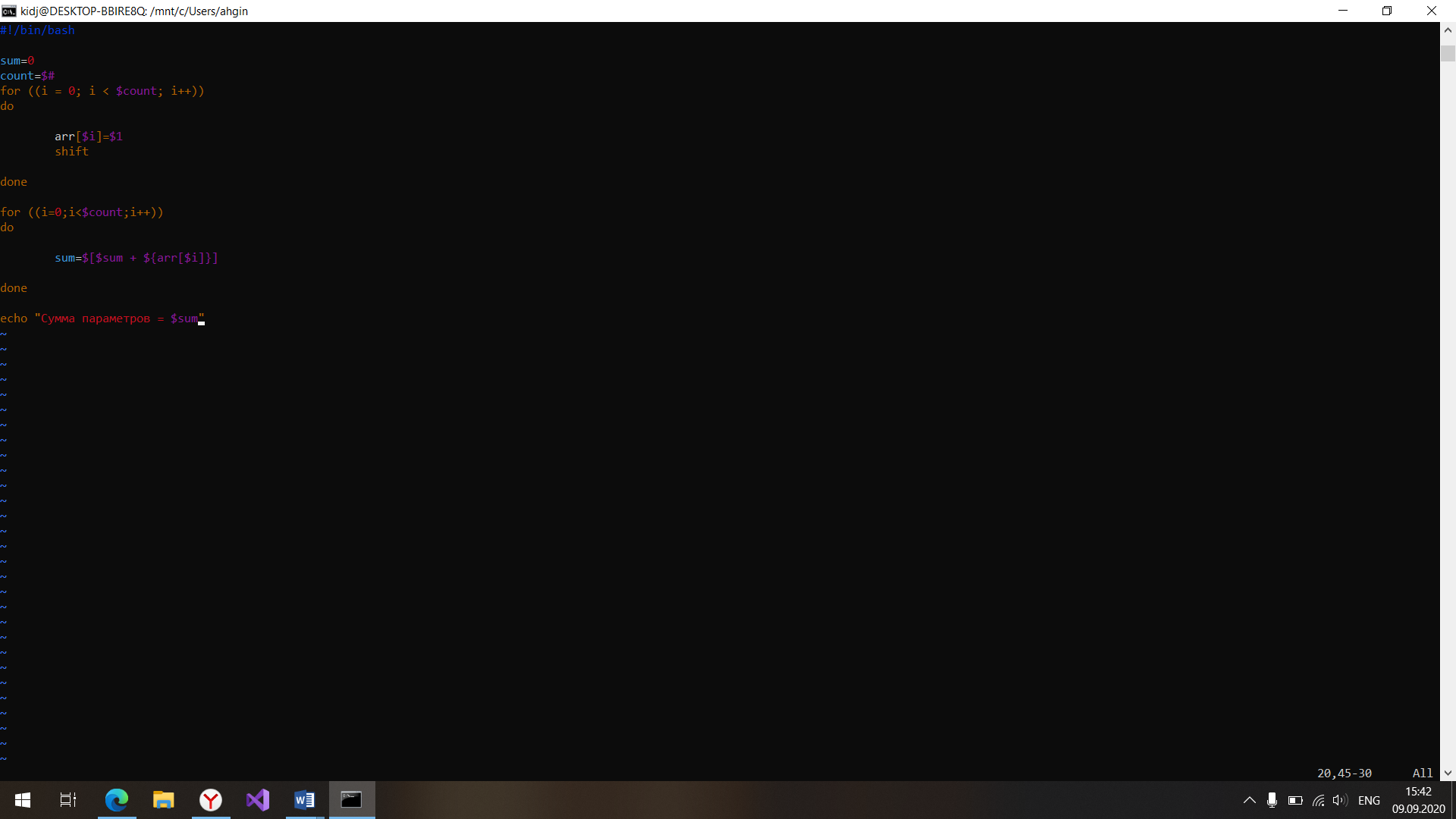
**for ((i=0;i<$count;i++))**

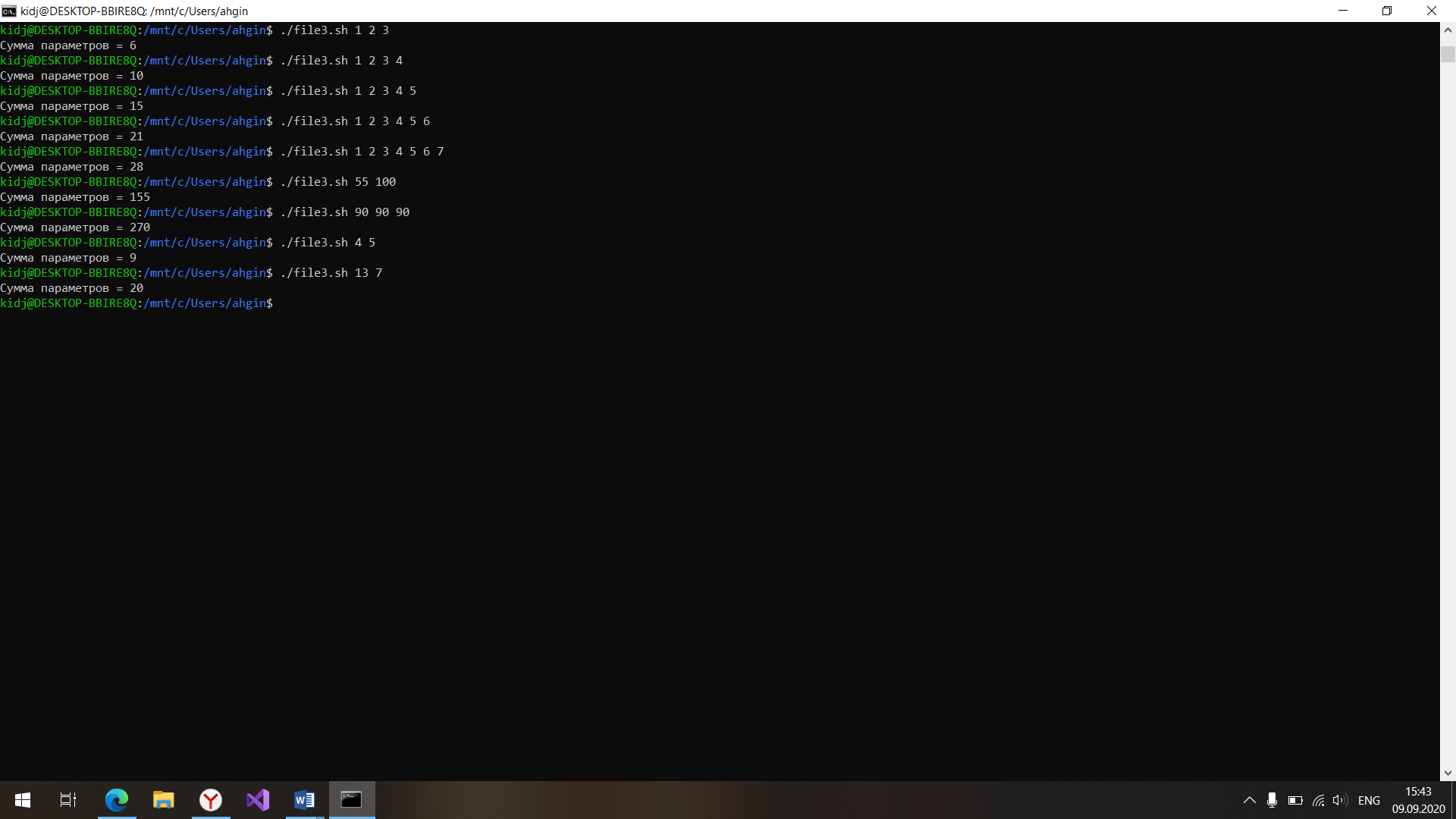
**do**

**sum=$[$sum + ${arr[$i]}]**

**done**

**echo "Сумма параметров = $sum"**



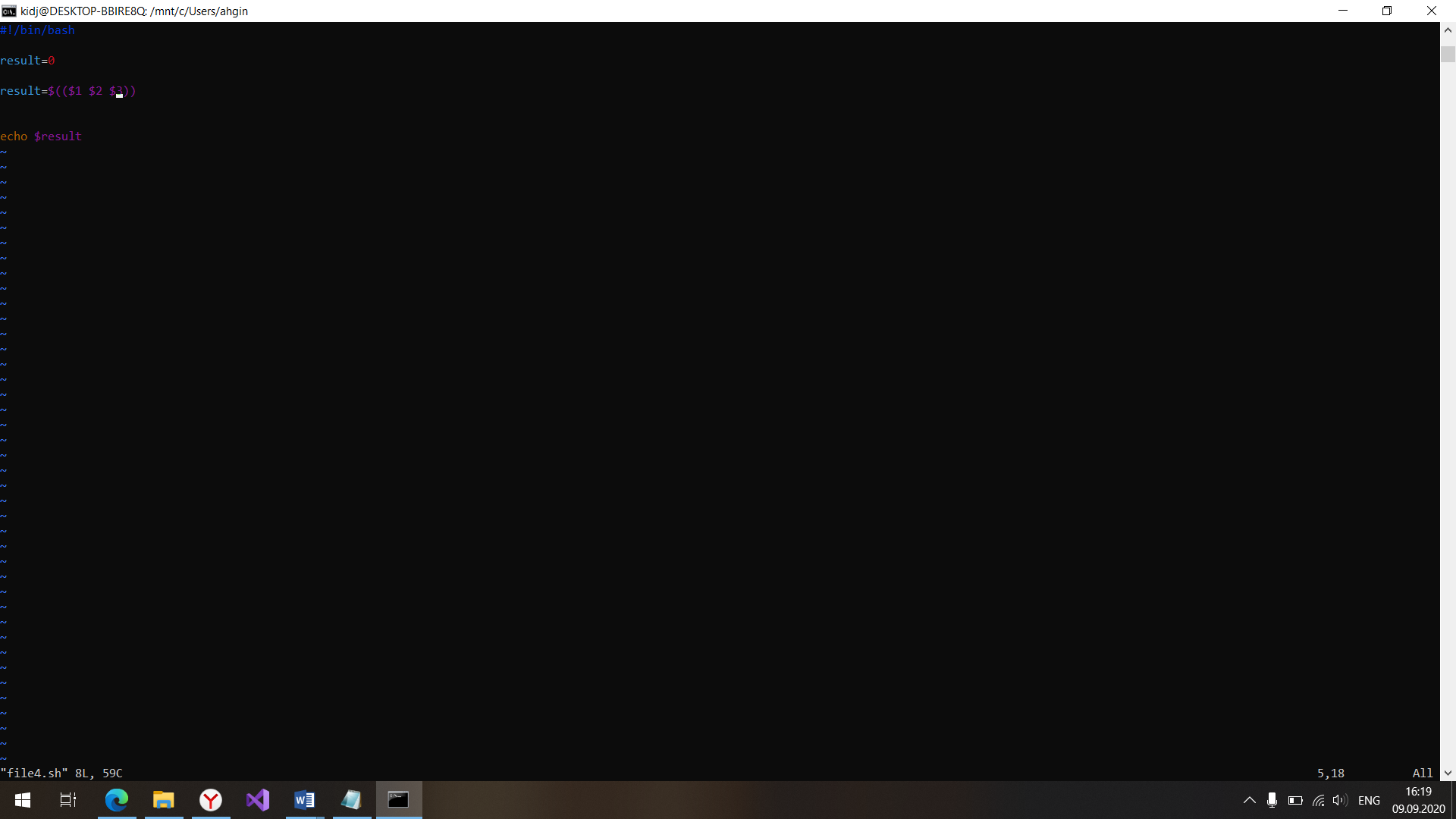


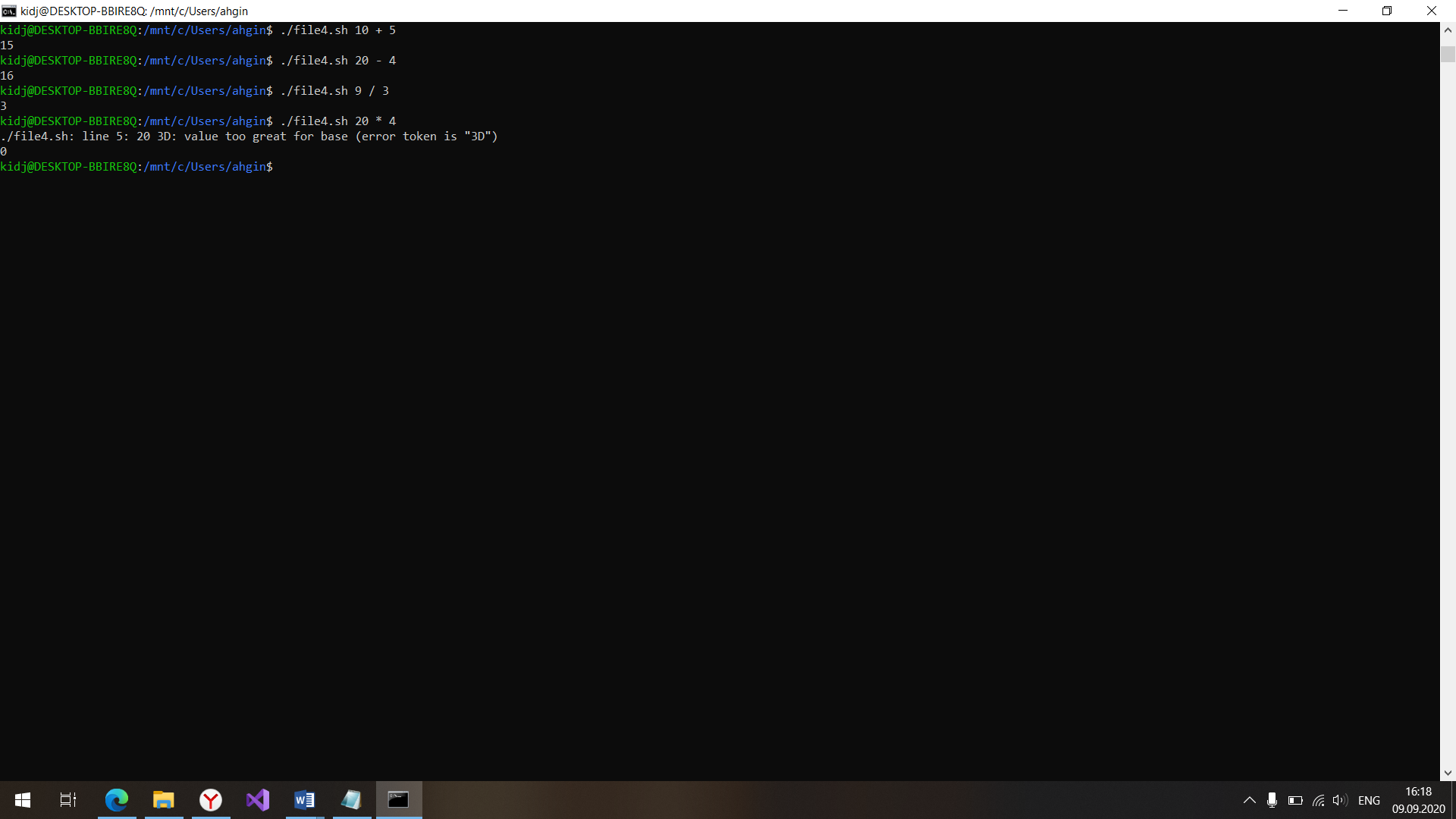
1. Написать сценарий, реализующий арифметический калькулятор. При запуске сценарию передаются три параметра, первым и третьим – числа, вторым – арифметическое действие (+, -, \*, /). На экран выводится результат этого арифметического действия для этих чисел.

**#!/bin/bash**

**result=$(($1 $2 $3))**

**echo $result**





1. Создать текстовой файл с написанными через пробел названиями. Написать сценарий, который создаст директории с указанными в файле названиями.

**#!/bin/bash**

**var=$(cat list.txt) # Заносим в переменную строку из файла**

**kol=$(echo $var|wc -w) # Считываем количество слов в файле**

**c=1**

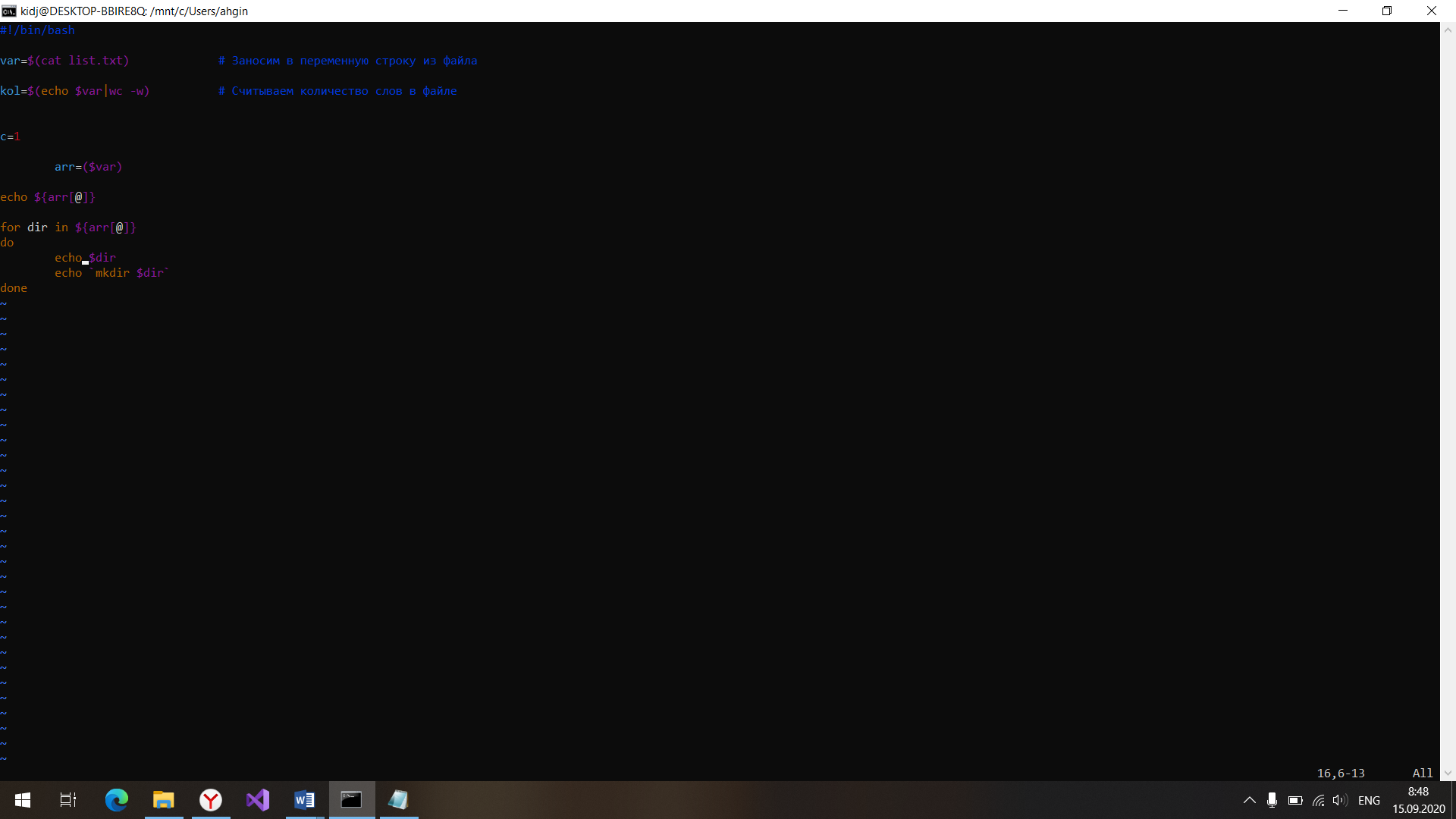
**arr=($var)**

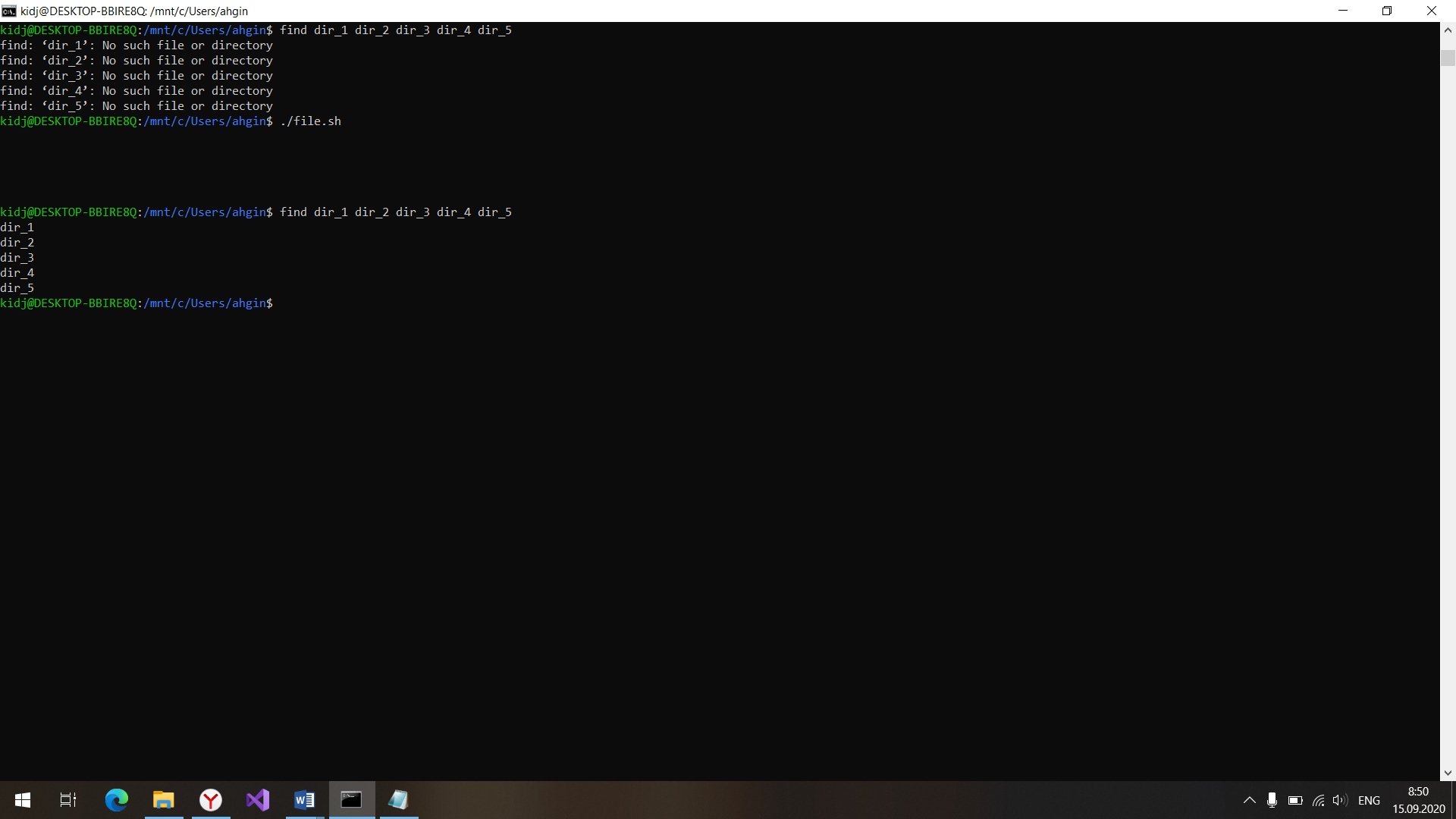
**for dir in ${arr[@]}**

**do**

**echo `mkdir $dir`**

**done**





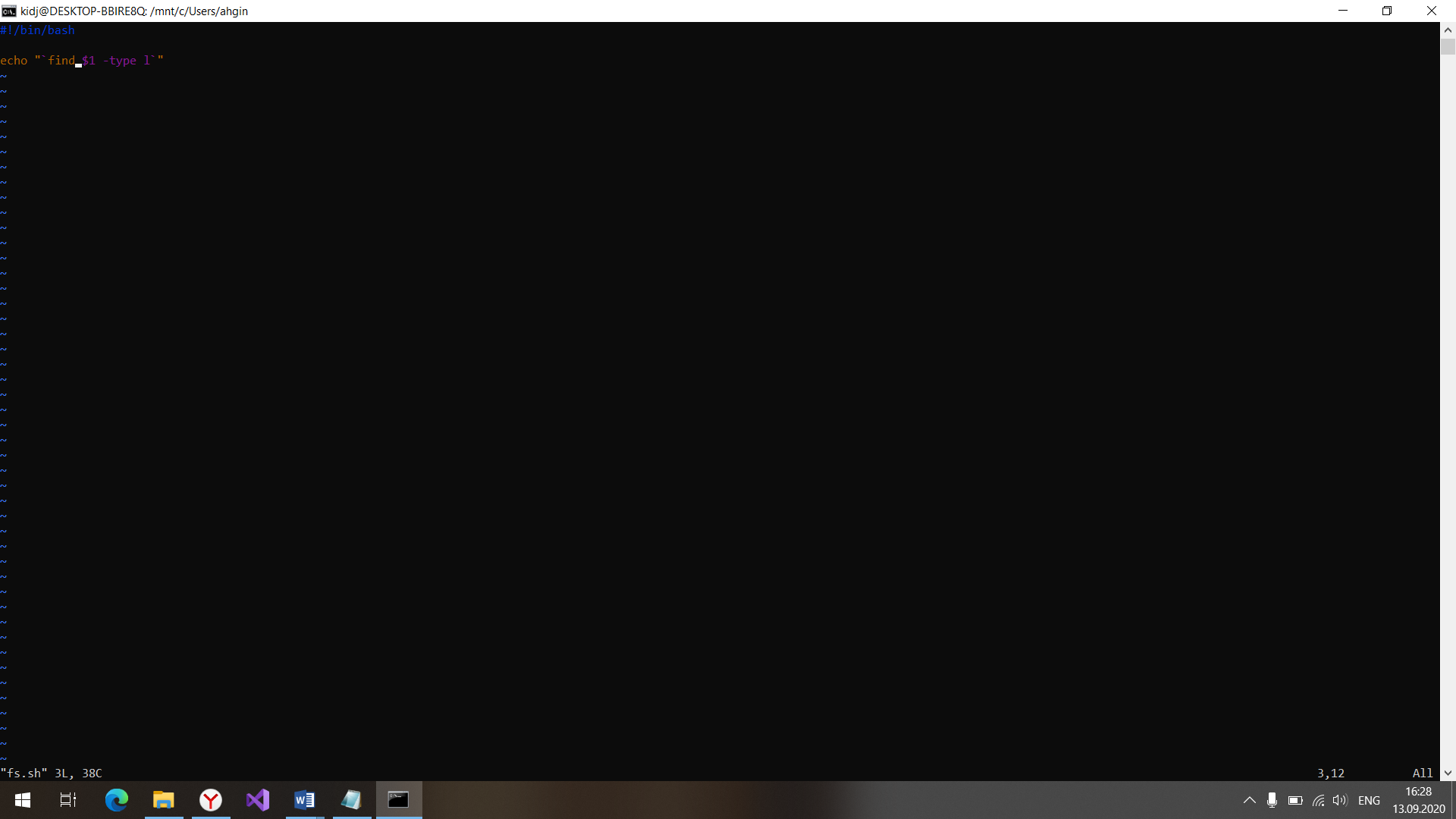
1. Сценарий принимает в качестве аргумента путь до директории и должен вывести количество символических ссылок, которые в ней находятся. Пример запуска такого сценария с именем scen.sh:

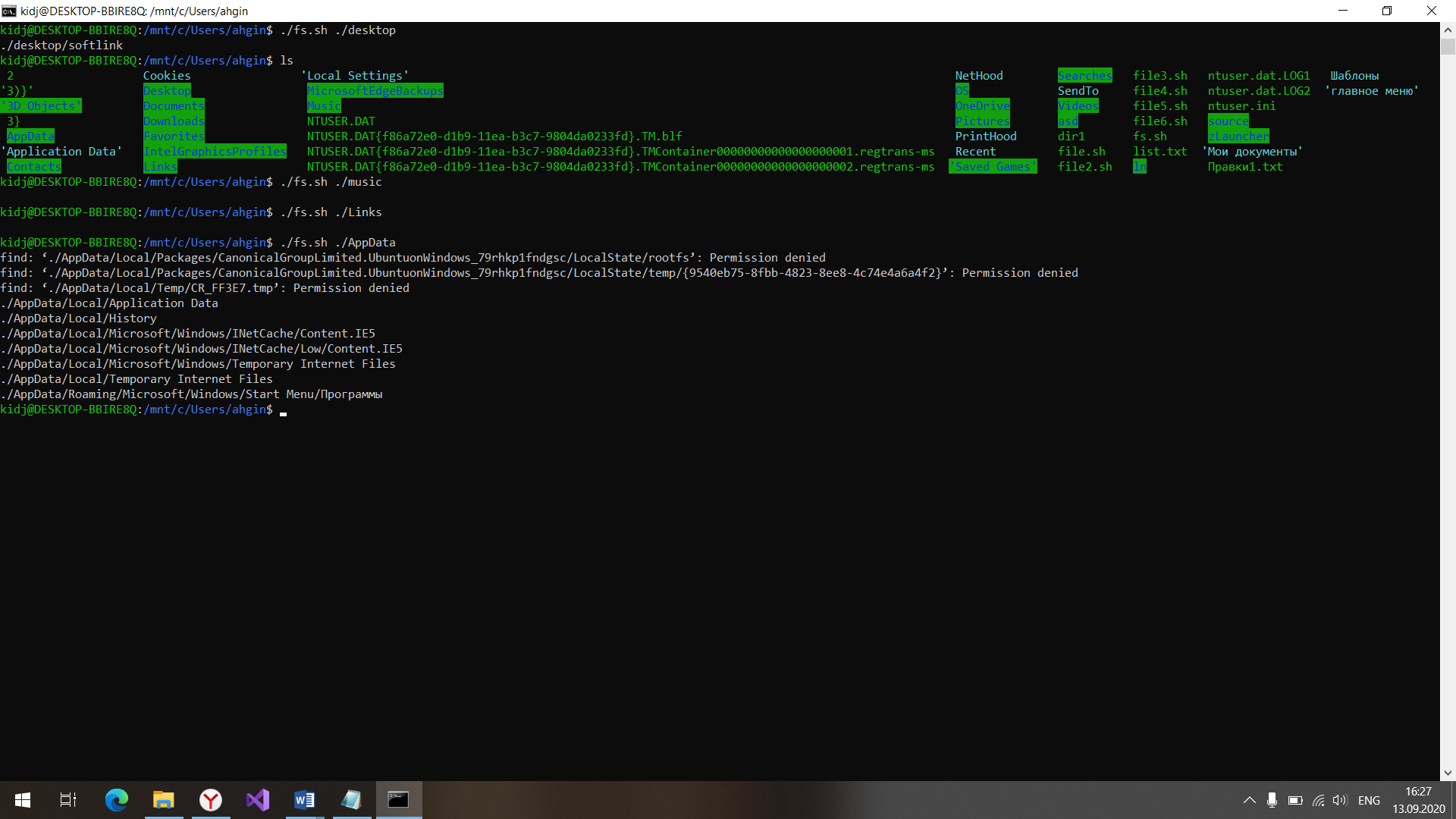
./scen.sh /home/stud/os

Организовать перебор содержимого директории можно при помощи цикла for

**#!/bin/bash**

**echo "`find $1 -type l|wc -w`"**





1. Создать сценарий, работающий с массивом. Он должен содержать три функции: show() – вывод массива на экран, push() – добавление элемента в конец массива, pop() – удаление последнего элемента в массиве. В функции push() значение добавляемого элемента запрашивается у пользователя для ввода с клавиатуры. При запуске сценария должно появляться меню, позволяющее пользователю по выбору выводить на экран массив, а также добавлять и удалять элементы.

**#!/bin/bash**

**push ()**

**{**

**read a**

**array+=($a)**

**}**

**show ()**

**{**

**echo "${array[@]}"**

**}**

**pop ()**

**{**

**unset array[-1]**

**}**

**while true**

**do**

**echo "Что сделать?"**

**echo "A. Вывести массив на экран"**

**echo "B. Добавить элемент массива"**

**echo "C. Удалить последний элемент массива"**

**echo "D. Выход"**

**read var**

**case $var in**

**"A"|"a")**

**echo -n "Ваш массив: "**

**show**

**echo ""**

**;;**

**"B"|"b")**

**echo -n "Добавление элемента: "**

**push**

**echo ""**

**;;**

**"C"|"c")**

**echo -n "Удаление последнего элемента: "**

**pop**

**echo ""**

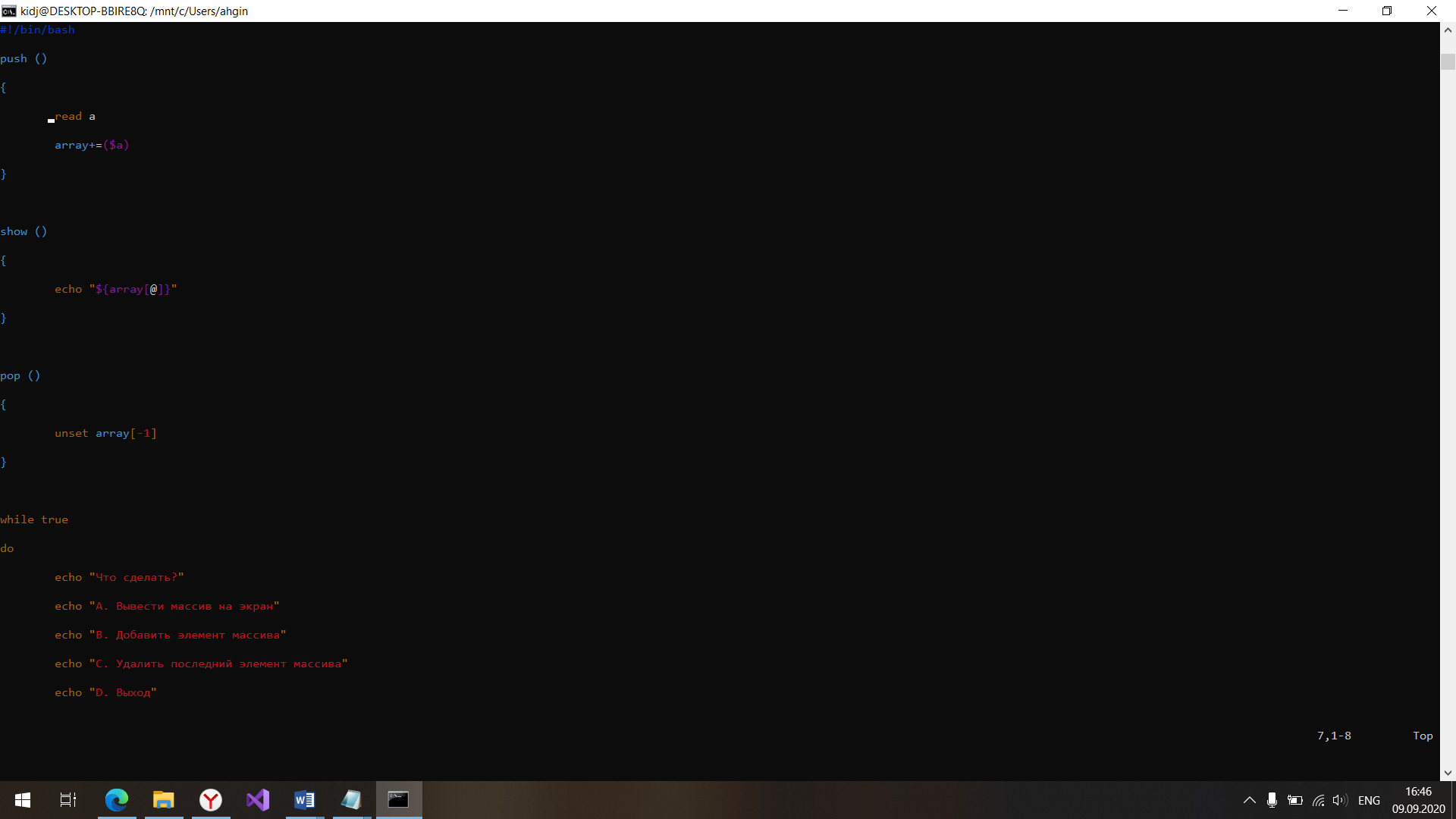
**;;**

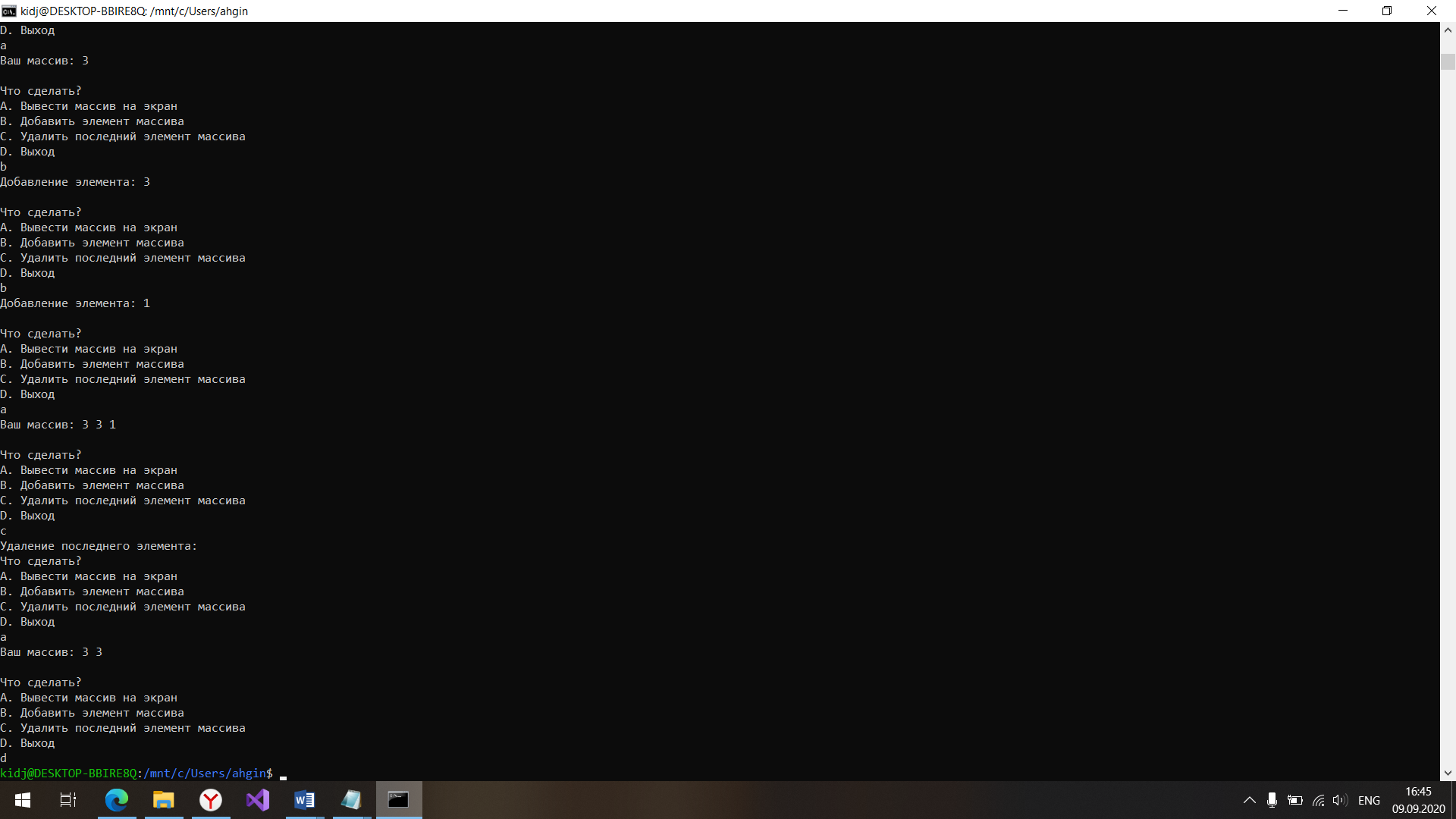
**"D"|"d")**

**break**

**esac**

**done**





# **Литература**

http://www.opennet.ru/docs/RUS/bash\_scripting\_guide/index.html