Лабораторная работа №6

Архитектура компьютера и операционные системы

Бабенко Константин, НКАбд-01-23

Содержание

[1 Цель работы 1](#_Toc145619377)

[2 Выполнение лабораторной работы 1](#_Toc145619378)

[2.1 Домашняя работа **Ошибка! Закладка не определена.**](#_Toc145619379)

[3 Выводы 7](#_Toc145619381)

# Цель работы

# Освоение арифметических инструкций языка ассемблера NASM.

# Выполнение лабораторной работы

# Символьные и численные данные в NASM

# 1) Создаю каталог для программ лабораторной работы № 6, перехожу в него и создаю файл lab6-1.asm:

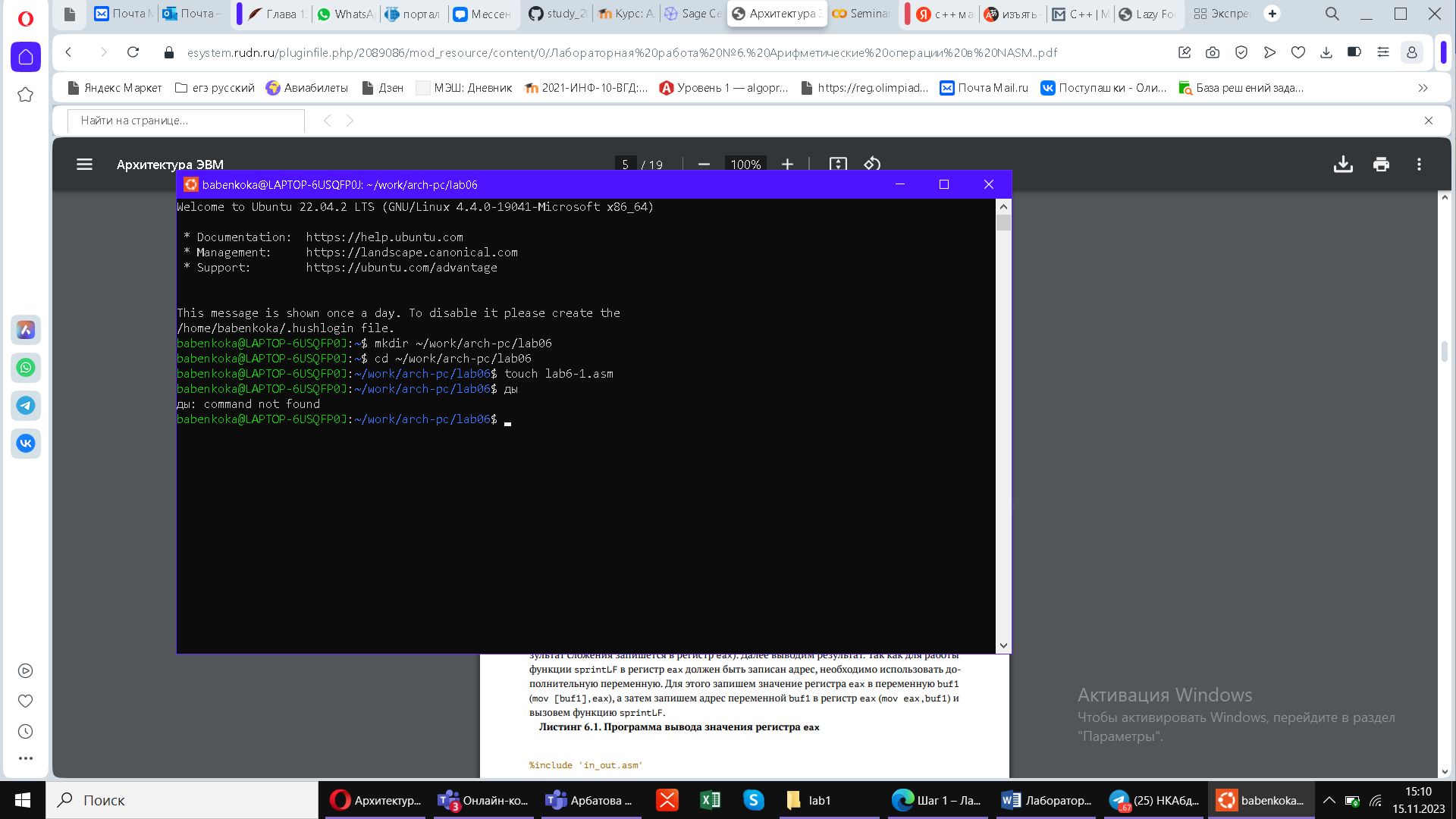


Figure 1: Создаю файл.

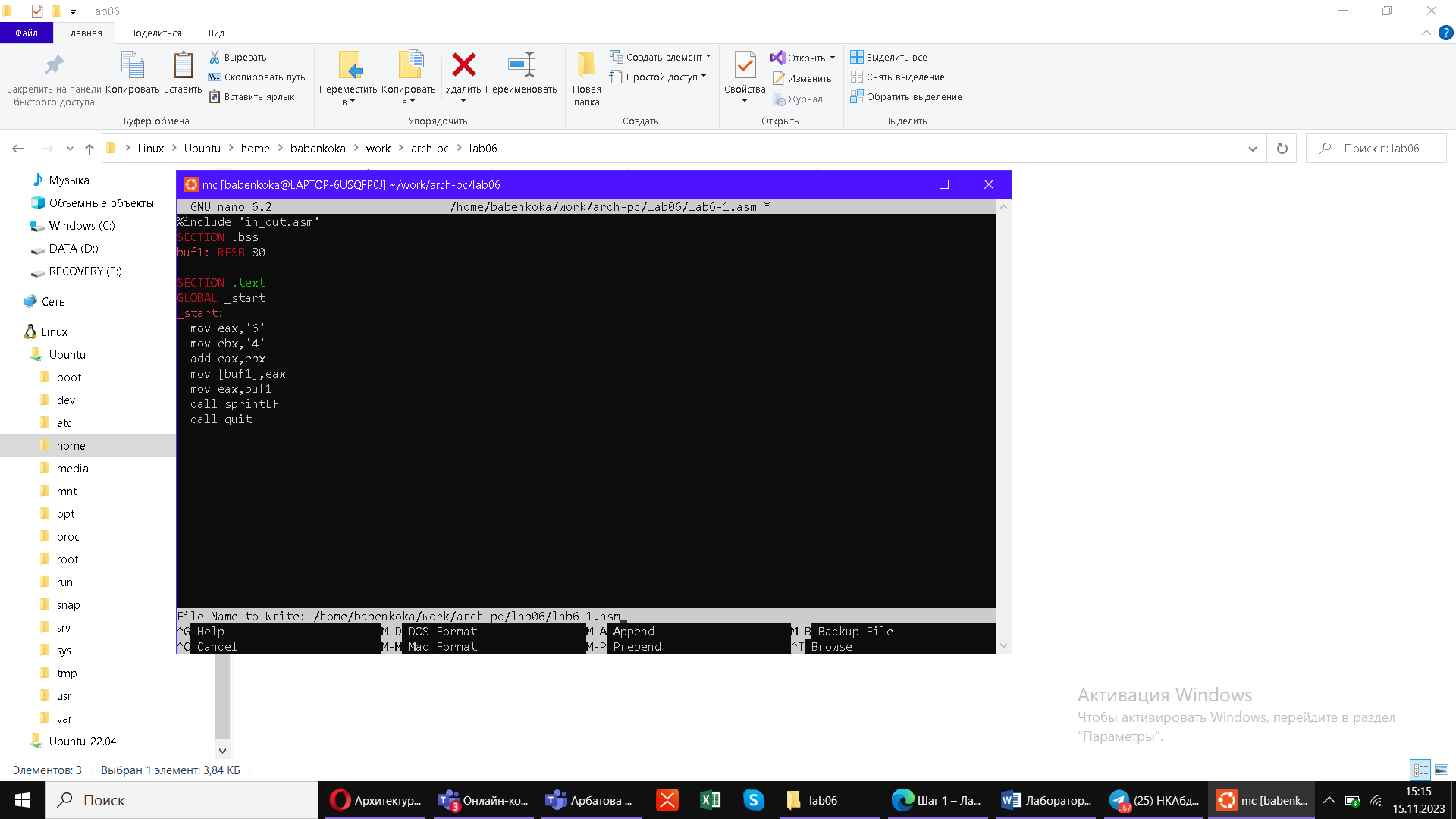
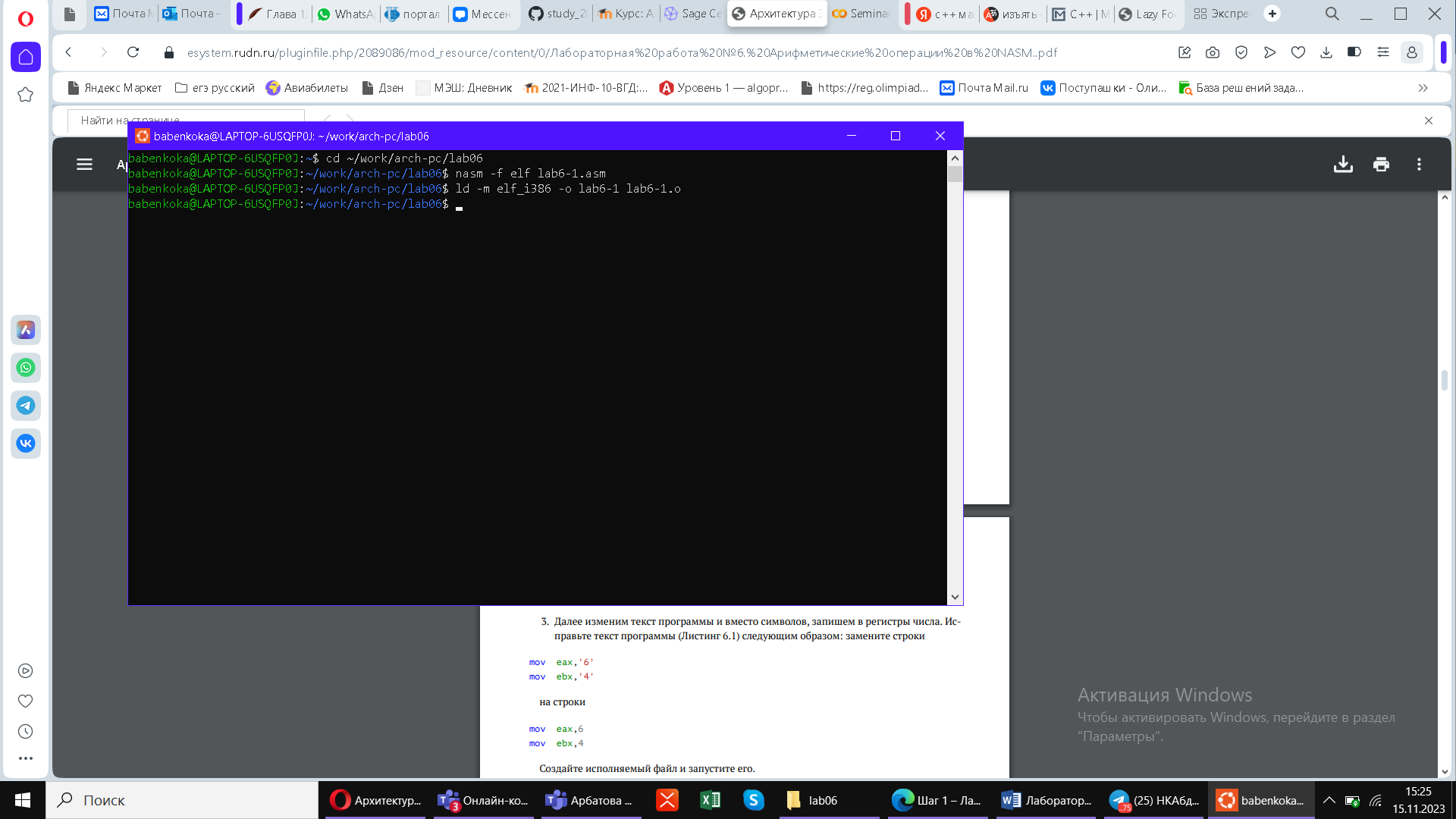
2) Ввожу в файл lab6-1.asm текст программы из листинга 6.1:

Figure 2: Ввожу текст программы.

Создаю исполняемый файл и запускаю его:



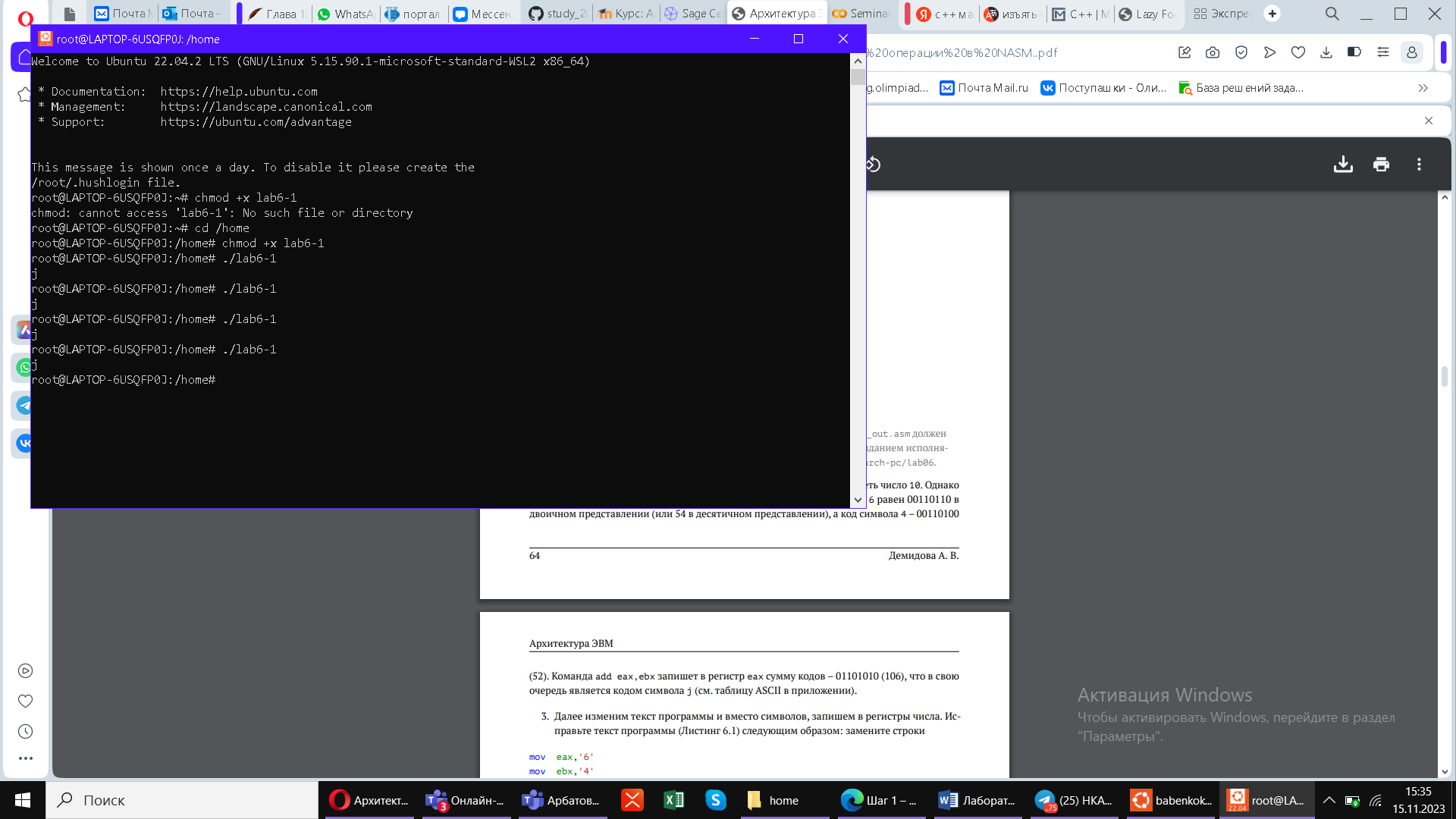


Figure 3: Работа файла.

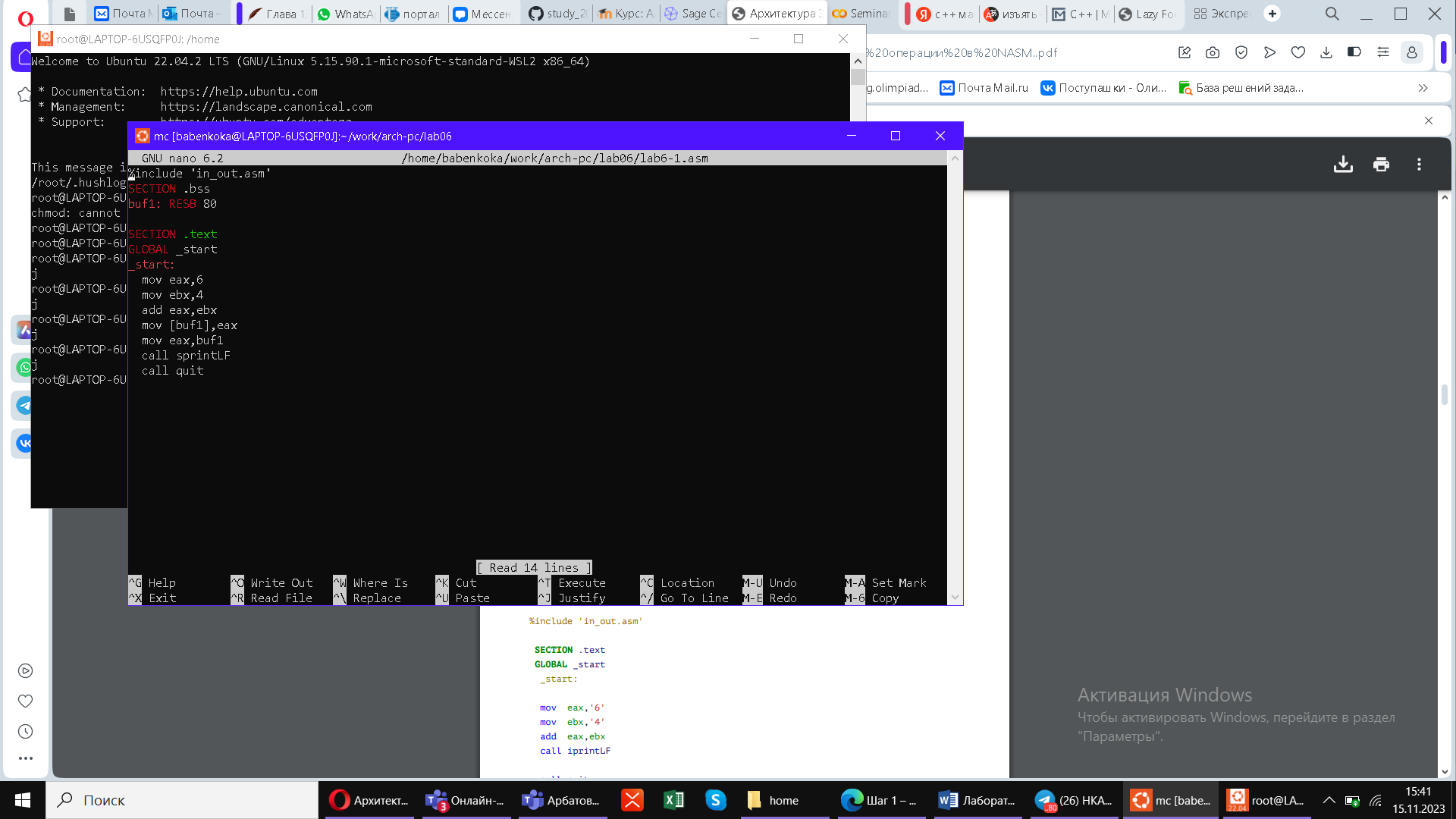
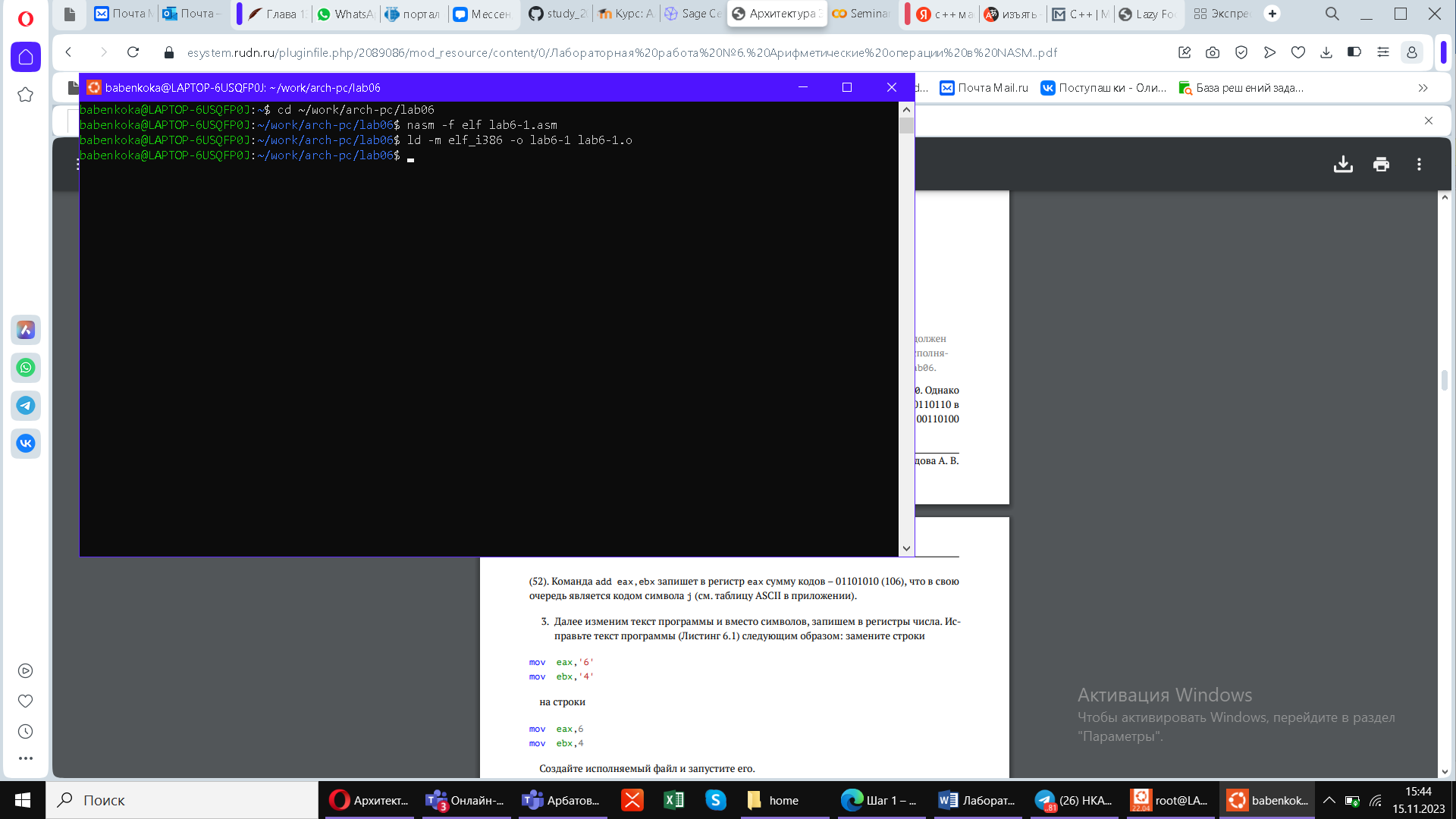
3) Далее изменяю текст программы и вместо символов, записываю в регистры числа. Исправляю текст программы следующим образом: заменить строки mov eax,'6' mov ebx,'4' на строки mov eax,6 mov ebx,4:

Figure 4: Измененный текст.

Создаю исполняемый файл и запустите его:



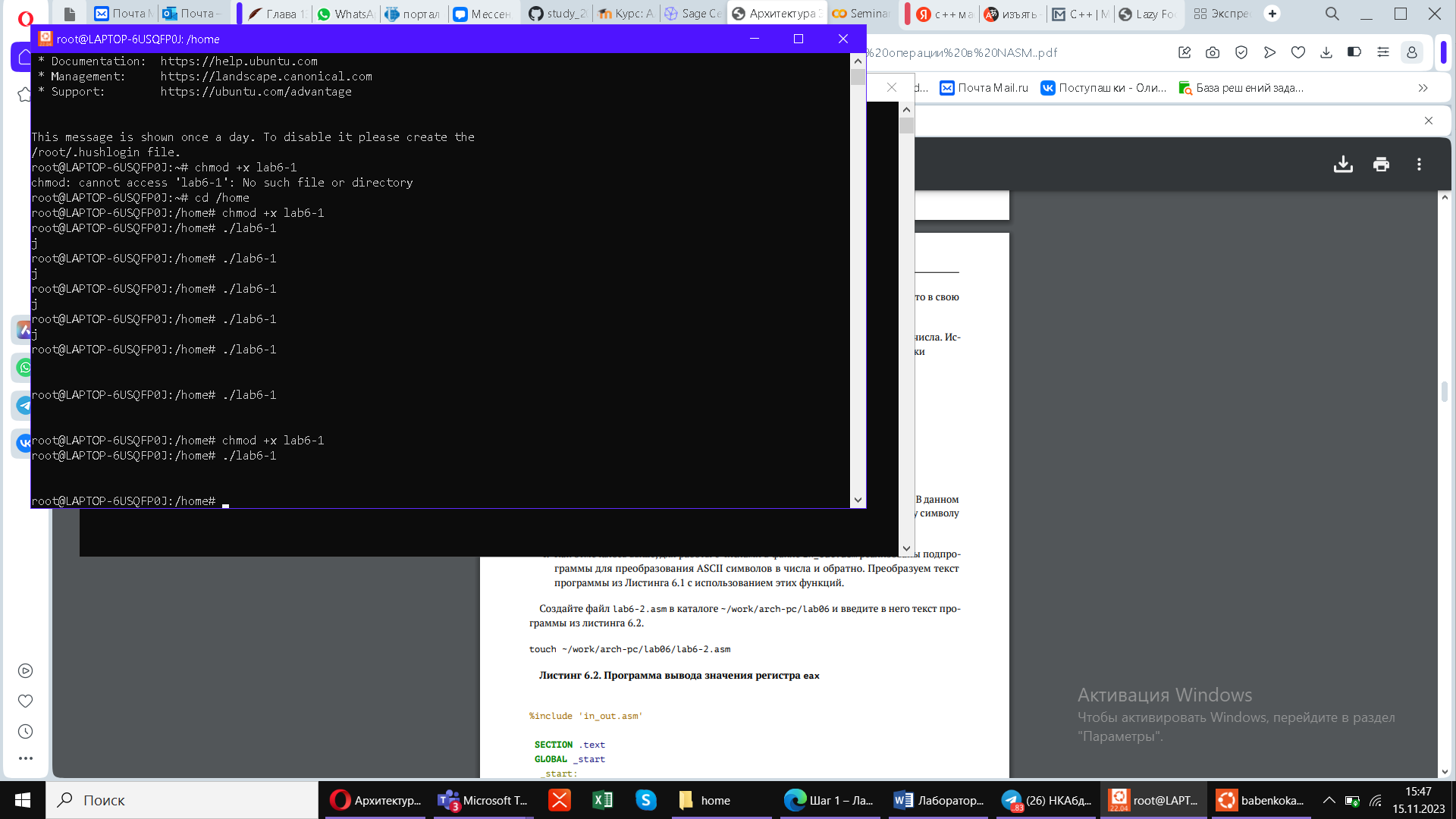


Figure 5: Работа измененного файла.

Код 10 соответствует символу BS, он не отображается.

4) Создаю файл lab6-2.asm в каталоге ~/work/arch-pc/lab06 и ввожу в него текст программы из листинга 6.2:

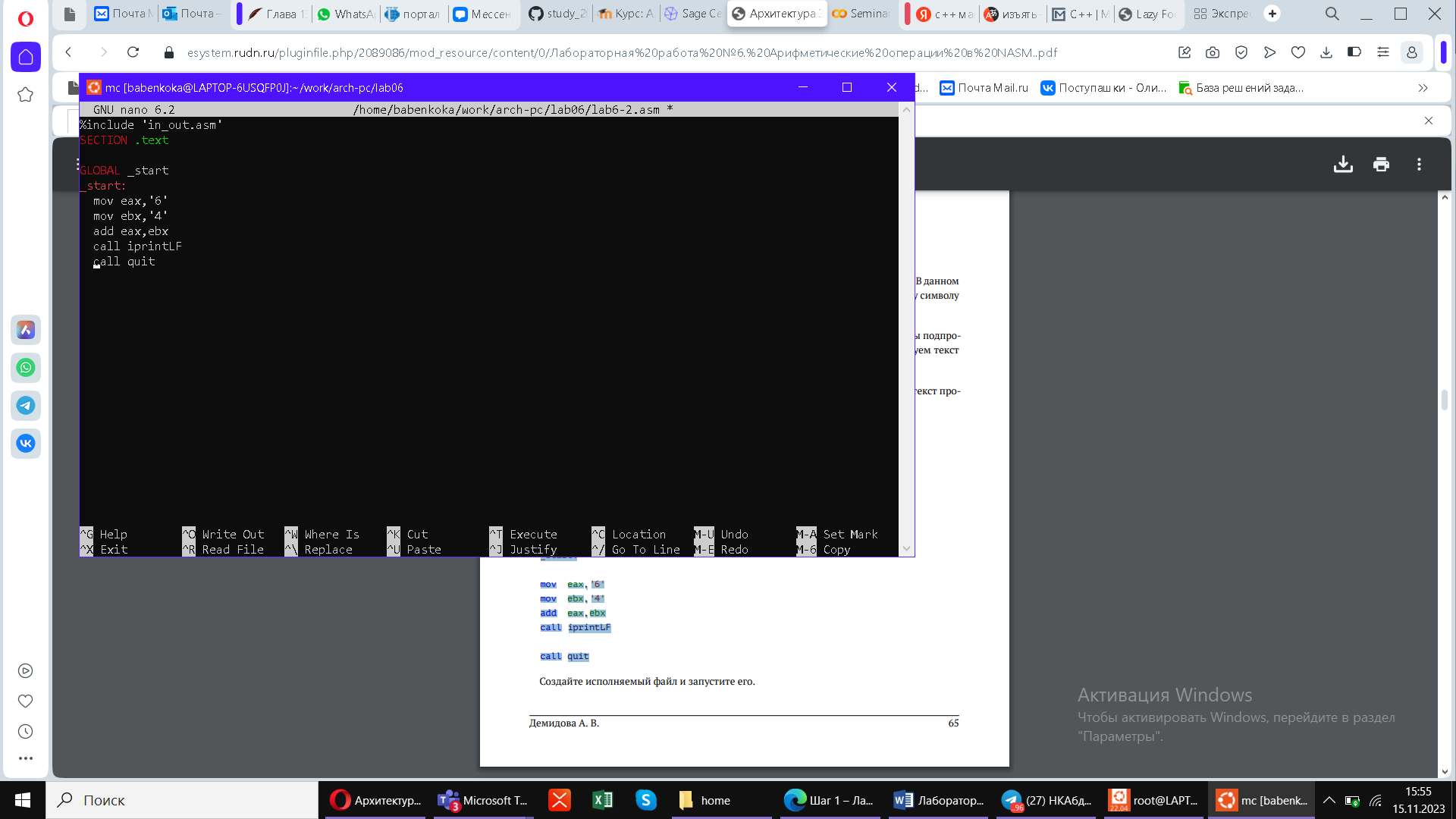
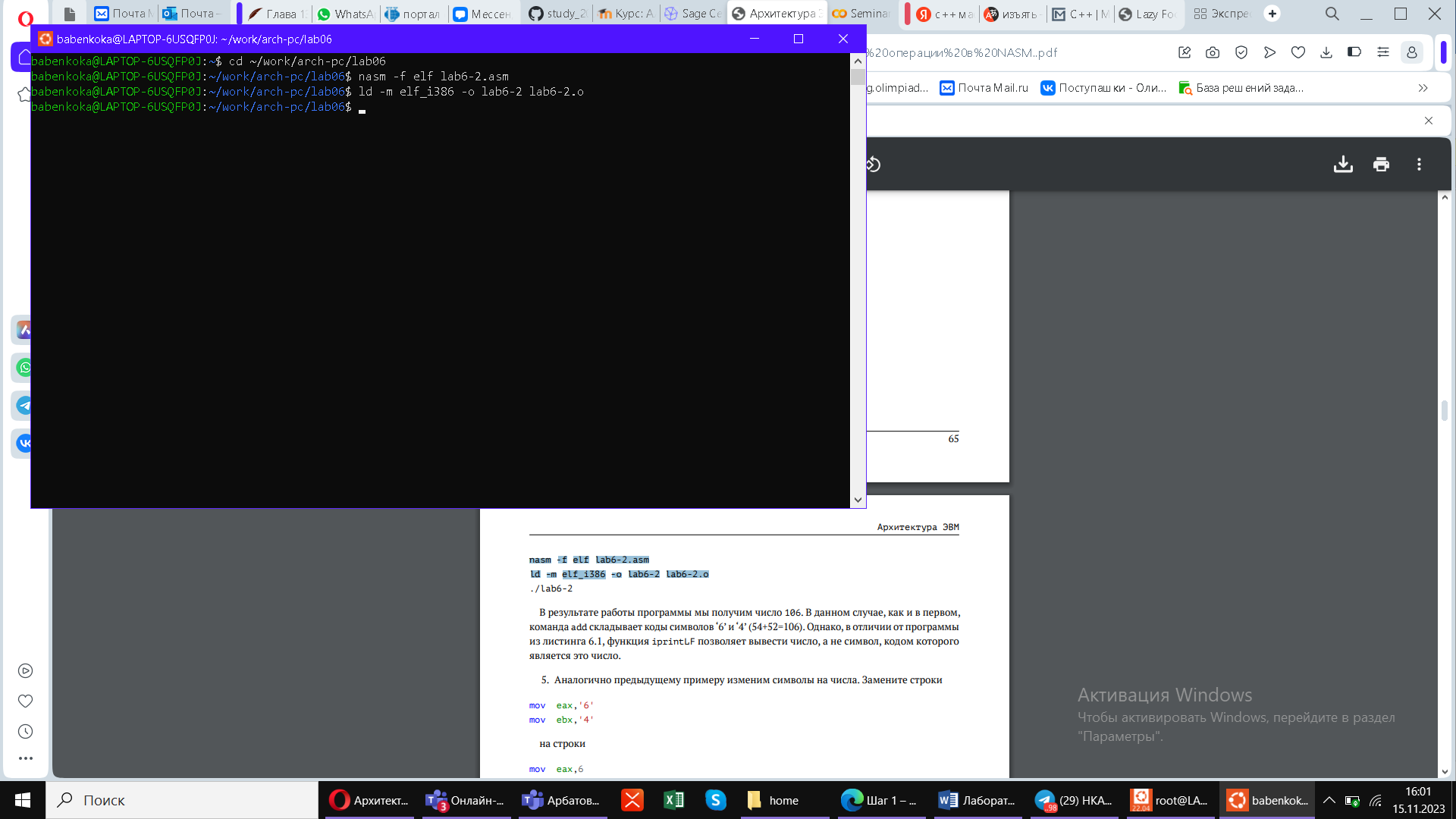


Figure 6: Введенный текст.

Создаю исполняемый файл и запускаю его:



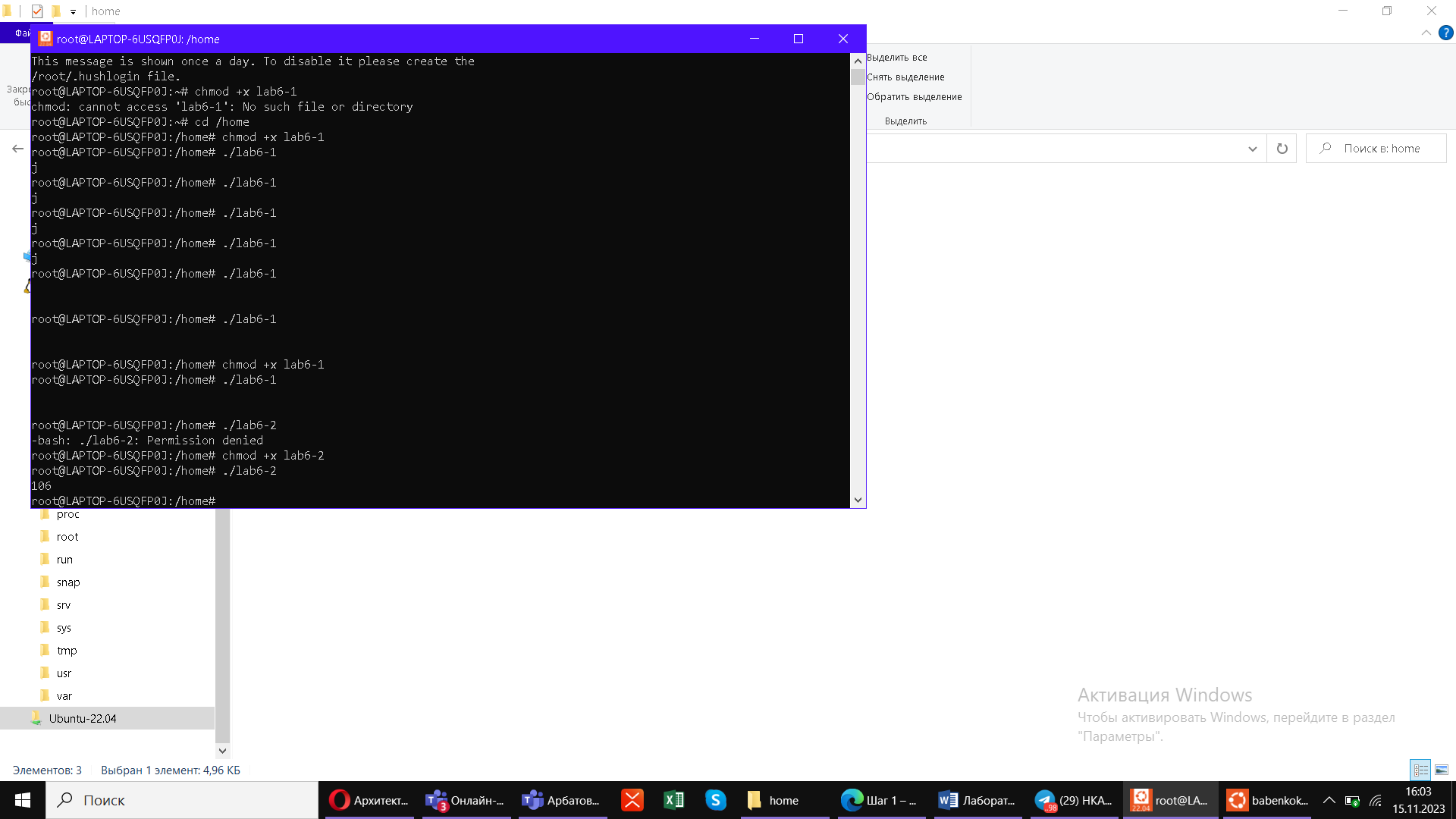


Figure 7: Работа второго файла.

В результате работы программы получено число 106. В данном случае, как и в первом, команда add складывает коды символов ‘6’ и ‘4’ (54+52=106). Однако, в отличии от программы lab6-1, функция iprintLF позволяет вывести число, а не символ, кодом которого является это число.

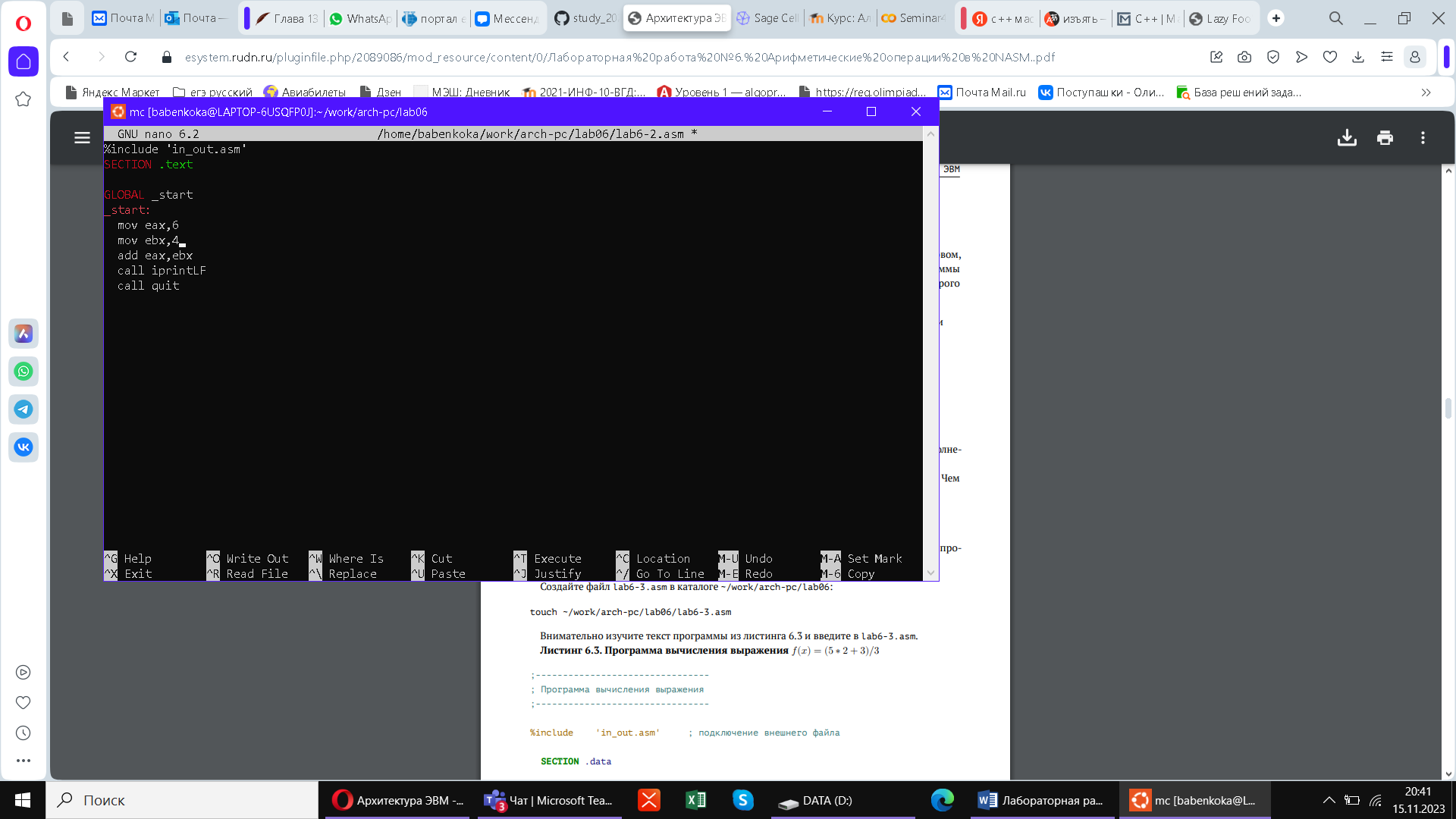
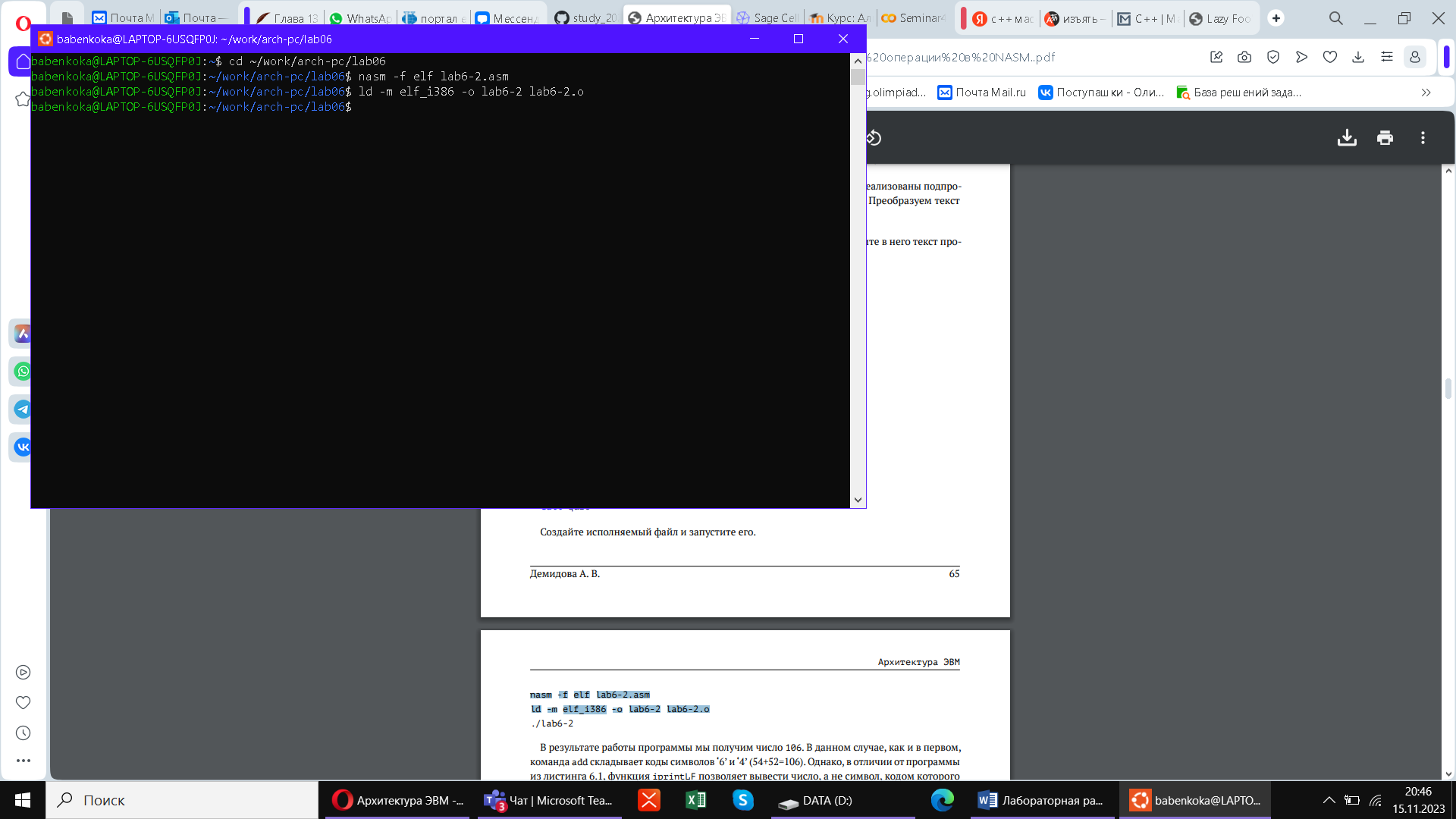
5) Аналогично предыдущему примеру изменяю символы на числа:

Figure 8: Изменения.

Создаю исполняемый файл и запускаю его:



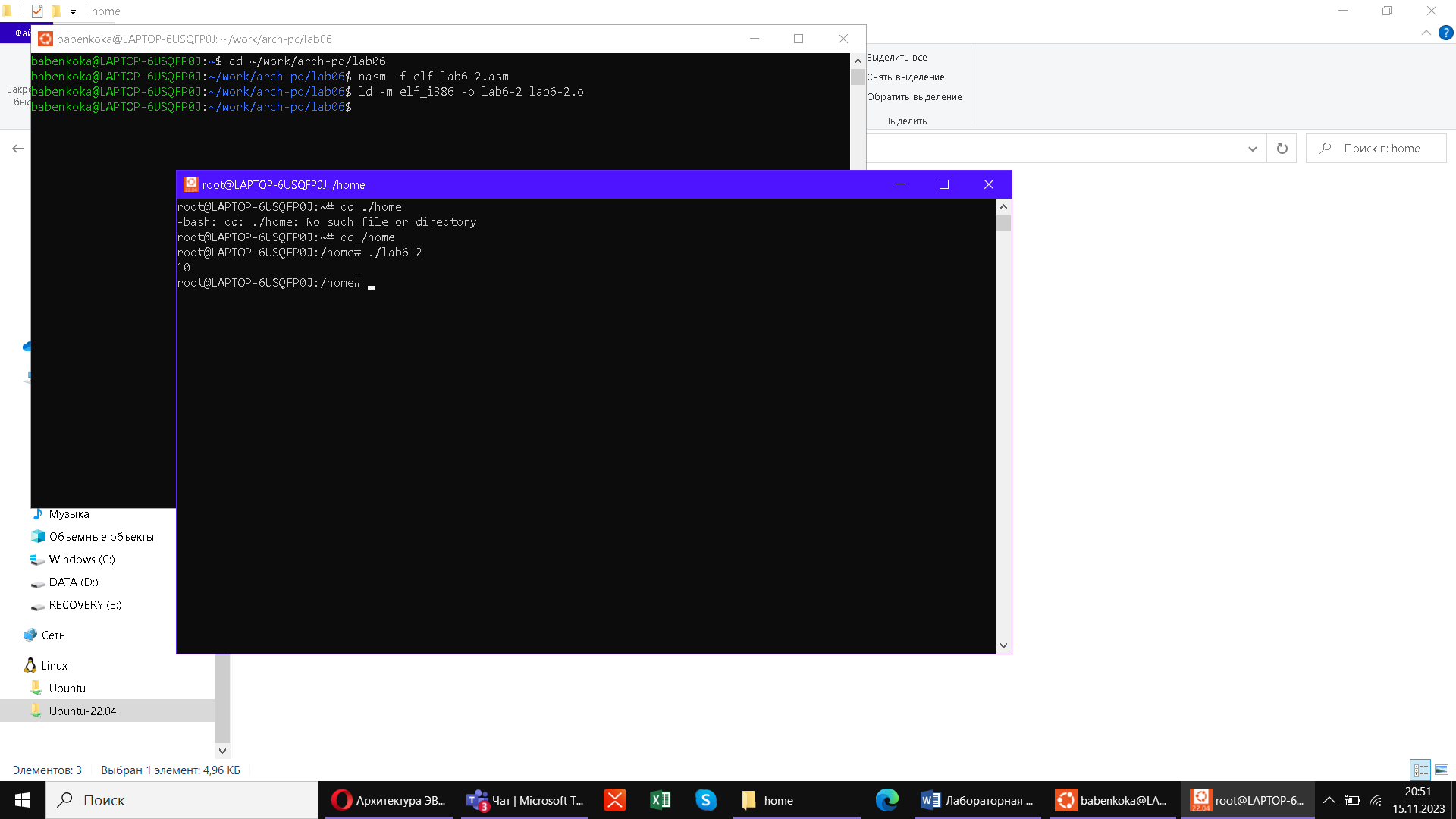


Figure 9: Работа измененной программы.

Заменяю функцию iprintLF на iprint. Создаю исполняемый файл и запускаю его:

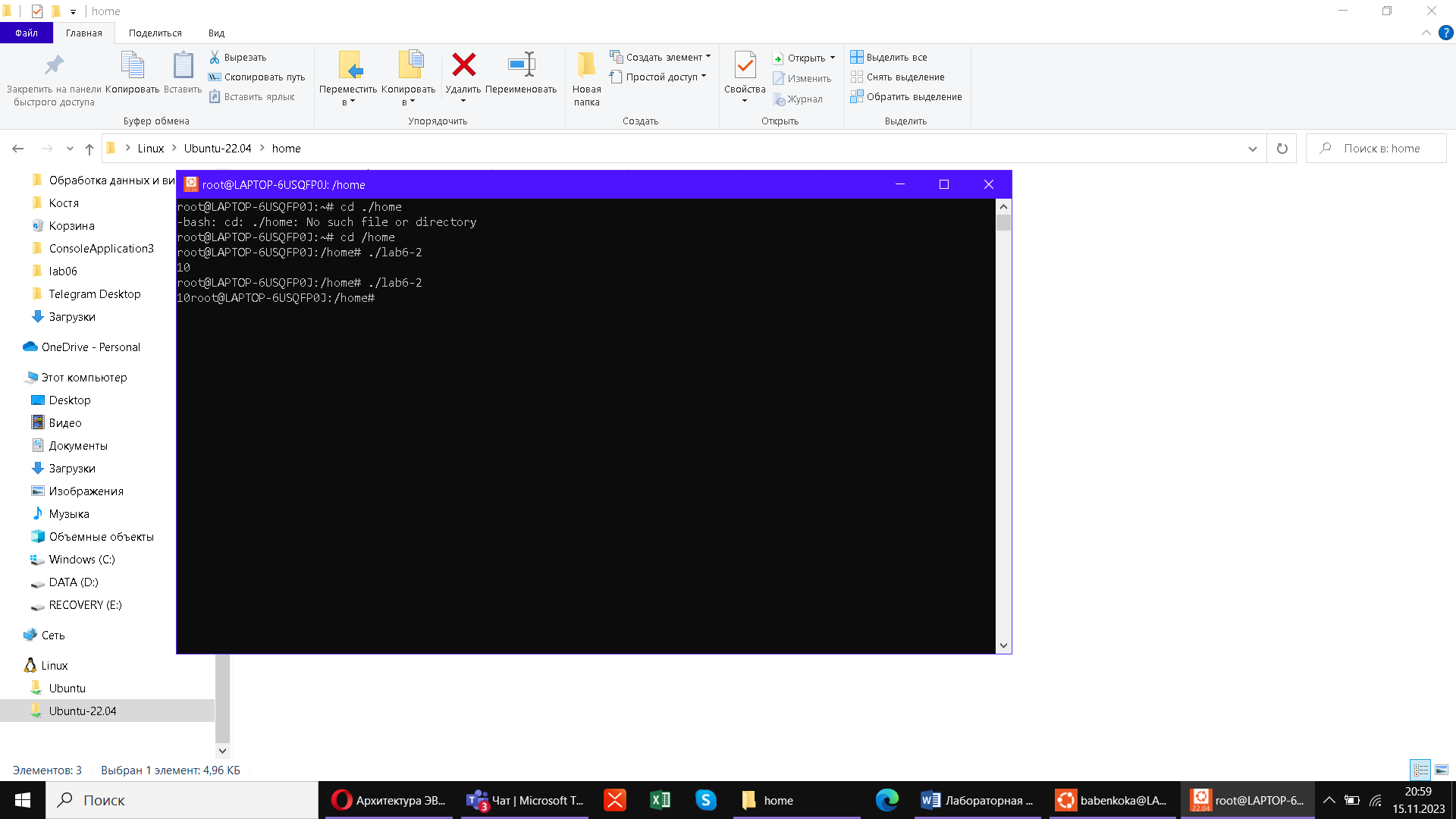


Figure 10: Работа программы без LF.

Функция iprintLF в отличие от iprint переводит курсор на другую строку.

# Выполнение арифметических операций в NASM

6) Создаю файл lab6-3.asm в каталоге ~/work/arch-pc/lab06:

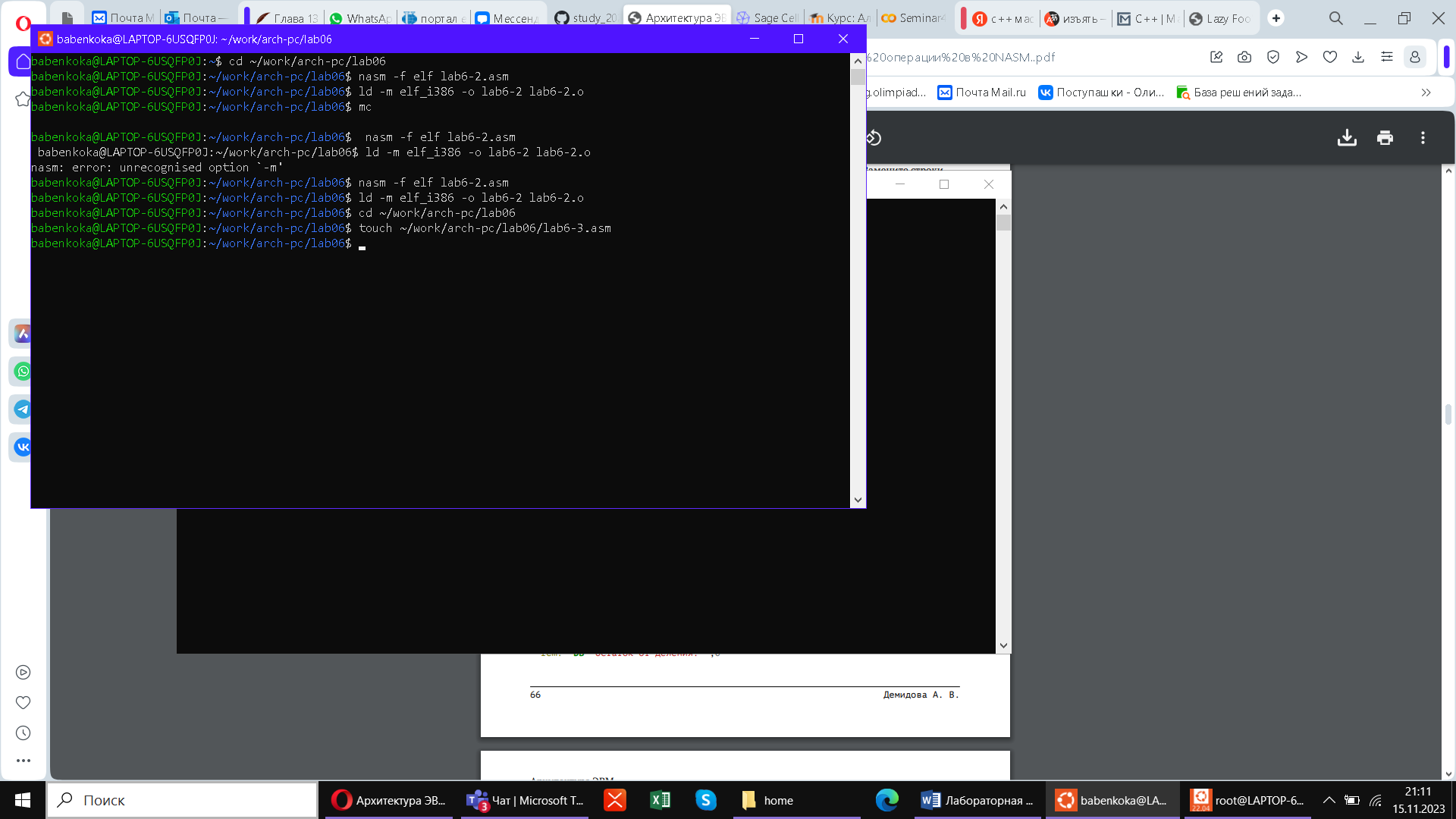


Figure 11: Программа.

Ввожу текст программы из листинга 6.3:

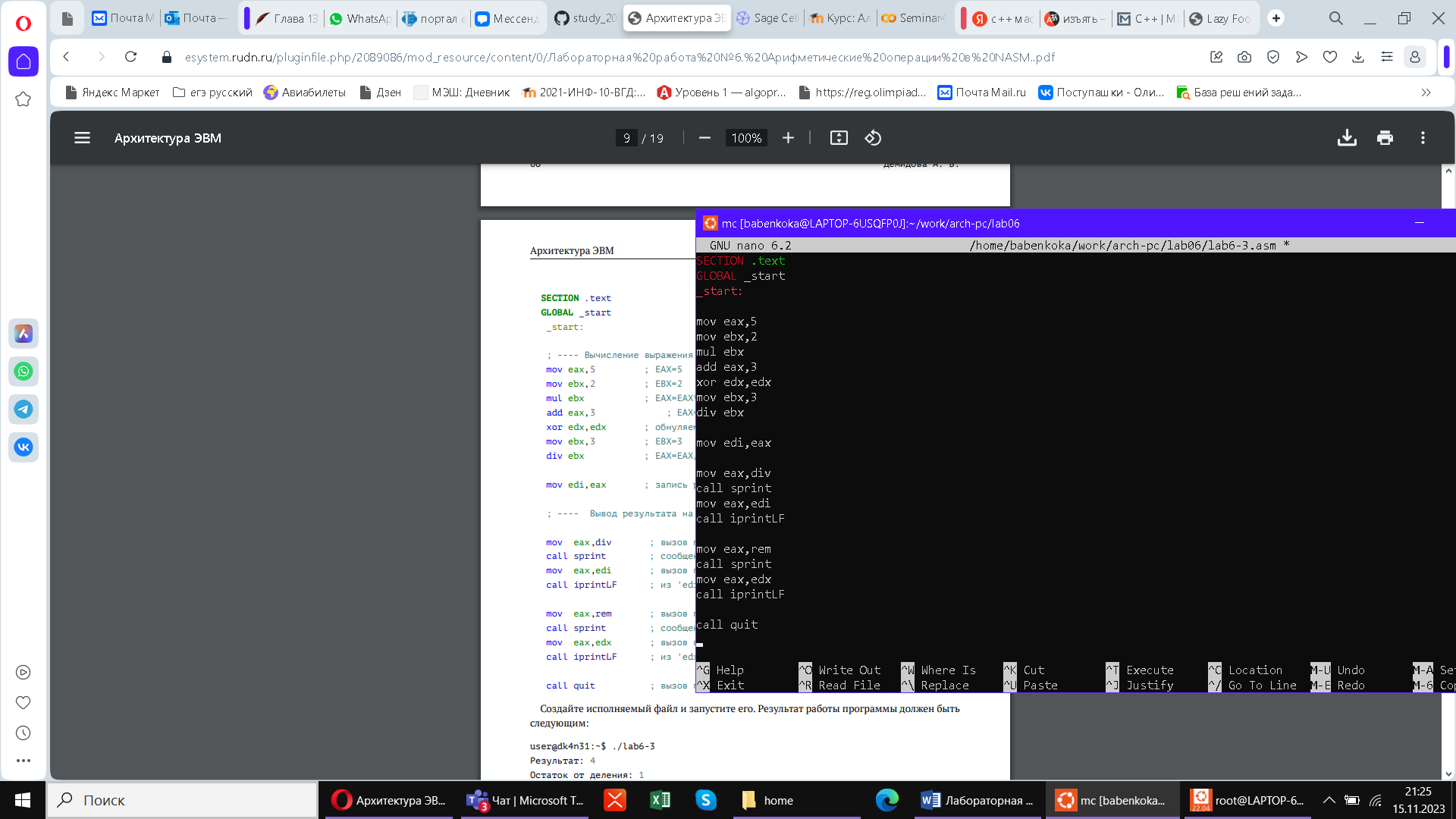
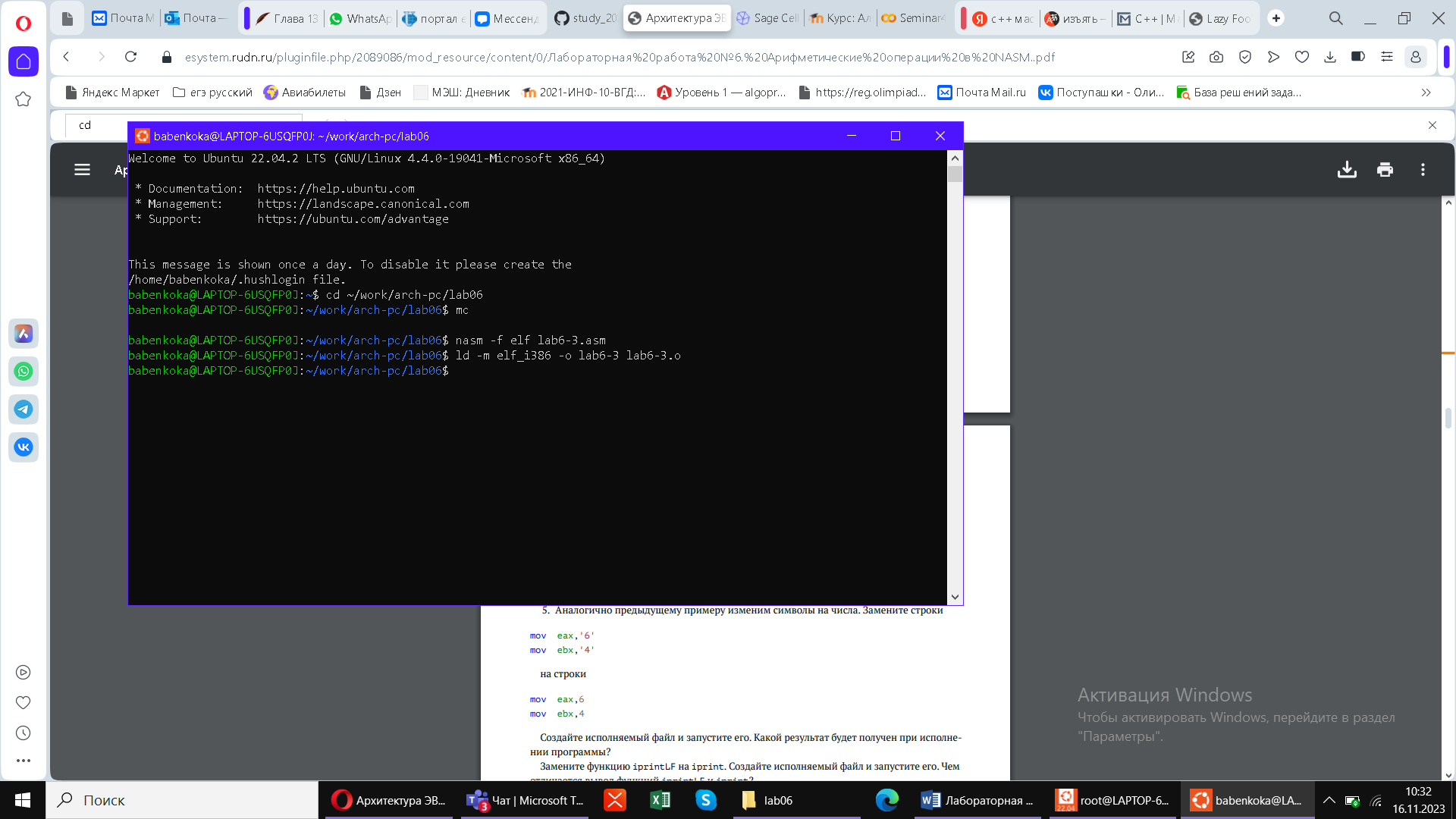


Figure 12: Введенный текст.

Создаю исполняемый файл и запускаю его:



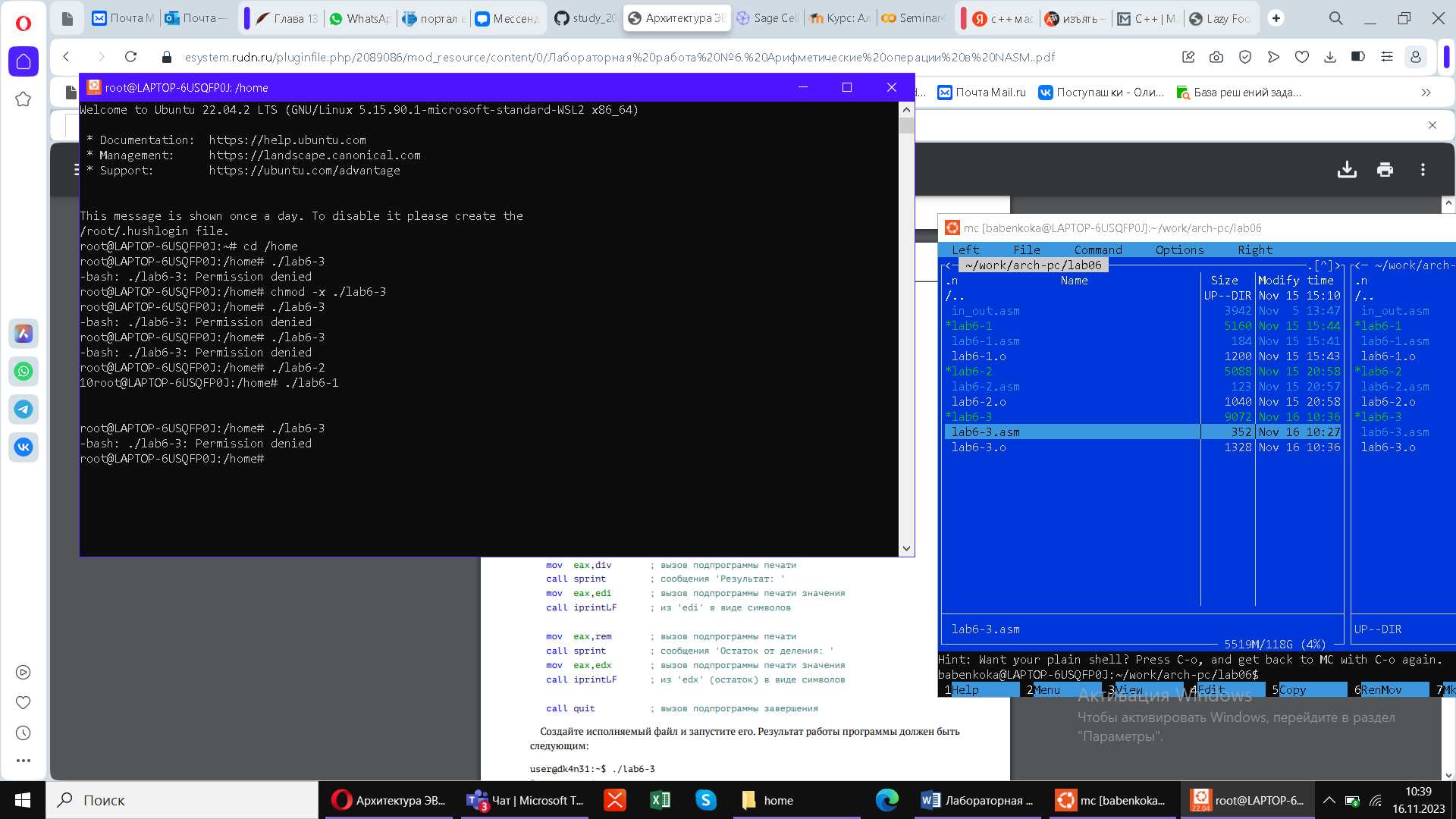


Figure 13: Работа программы.

7) Создаю файл variant.asm в каталоге ~/work/arch-pc/lab06:

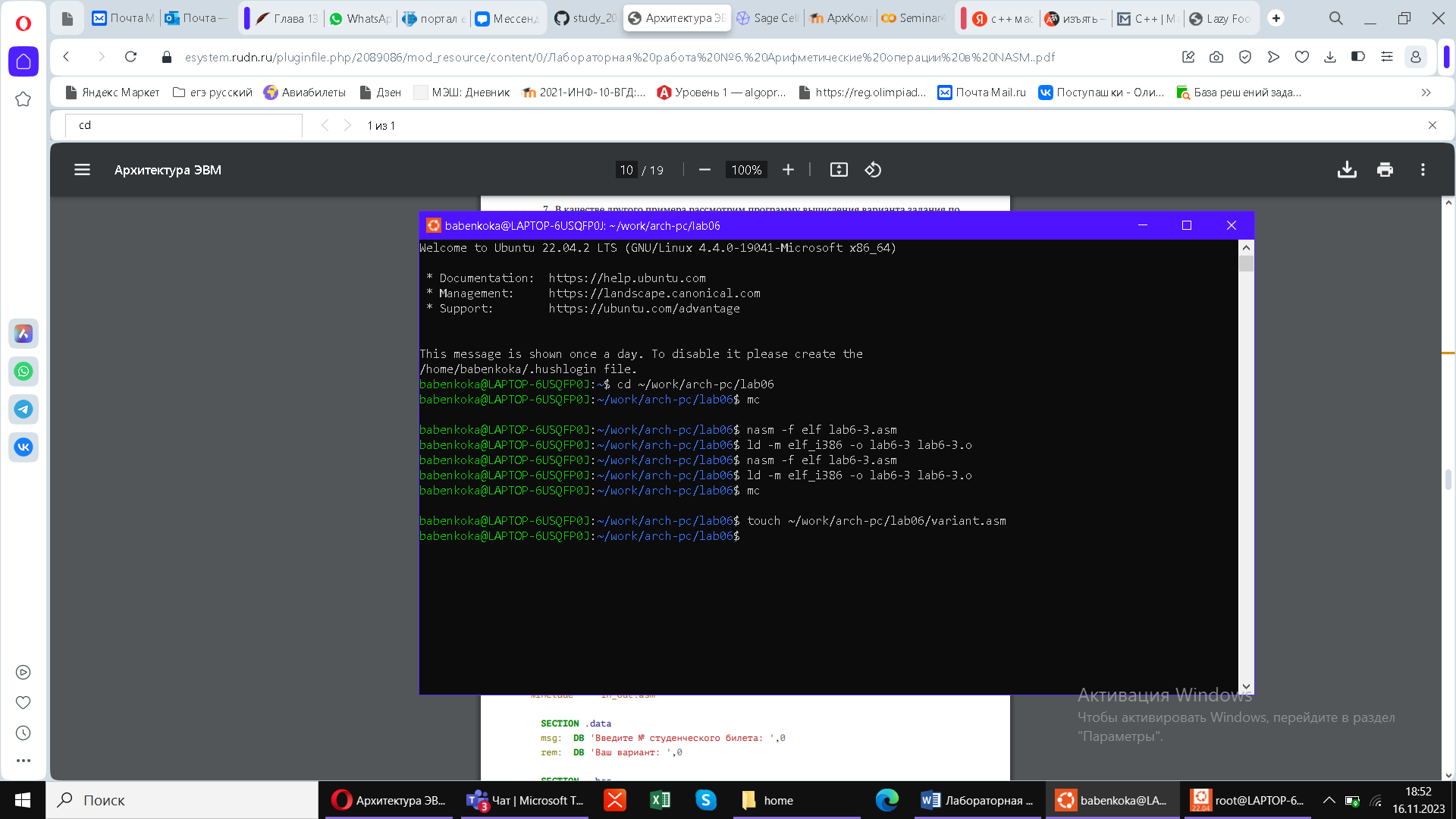


Figure 14: Создаю новый файл.

Ввожу в файл variant.asm текст программы из листинга 6.4:

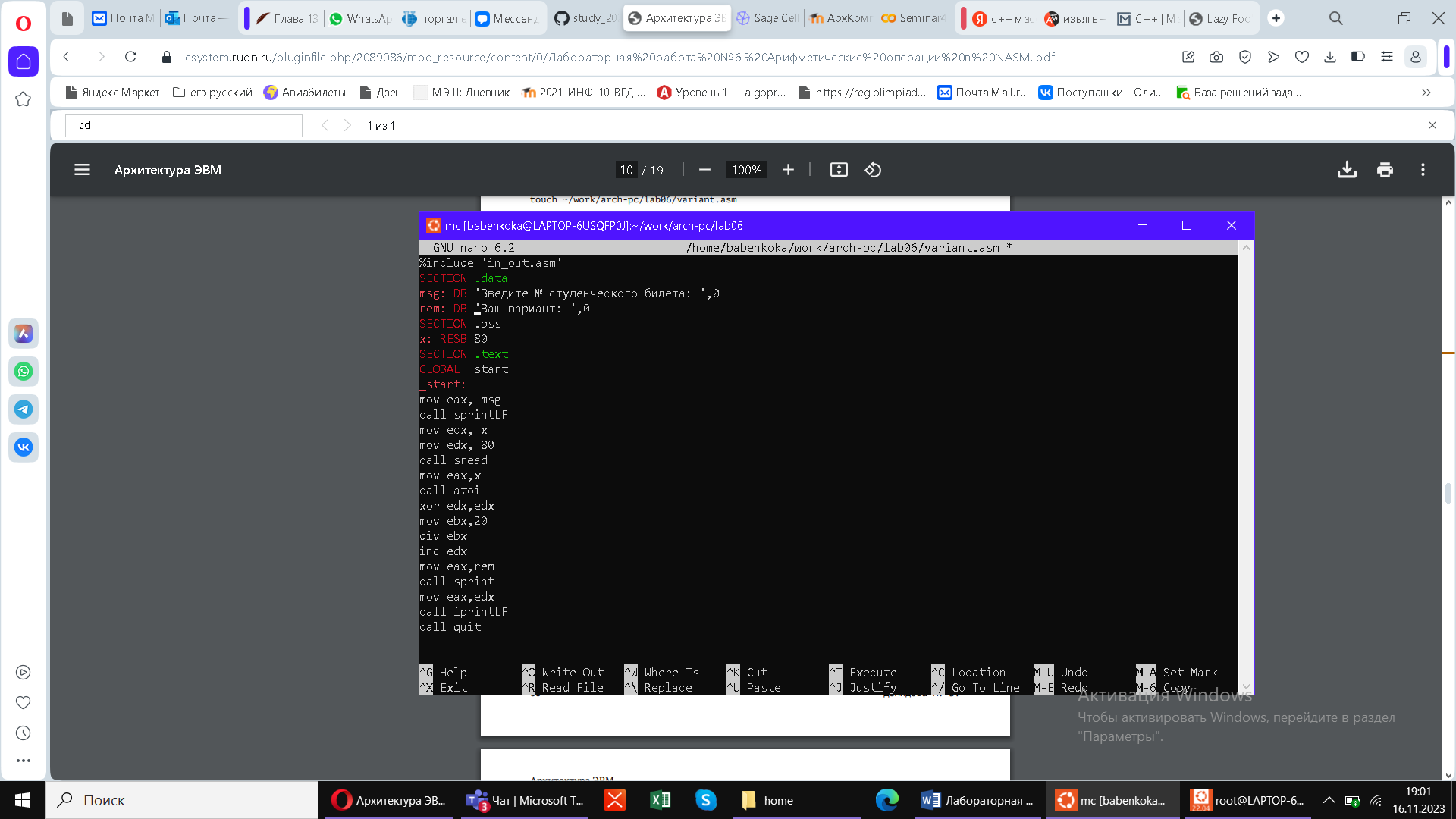


Figure 15: Текст программы.

1. Строки, отвечающие за вывод на экран сообщения «Ваш вариант:»:

mov eax,rem

call sprint

mov eax,edx

call iprintLF

1. Следующие инструкции отвечают за считывание данных:

mov ecx, x

mov edx, 80

call sread

1. Инструкция “call atoi” используется для превращения ASCII кода в число.
2. За вычисления варианта отвечают:

xor edx,edx

mov ebx,20

div ebx

inc edx

1. В edx.
2. Инструкция “inc edx” используется для прибавления единицы.
3. За вывод на экран результата вычислений отвечают:

mov eax,edx

call iprintLF

# Домашняя работа

Ввожу программу вычисления выражения 𝑦 = (10 + 2𝑥)/3:

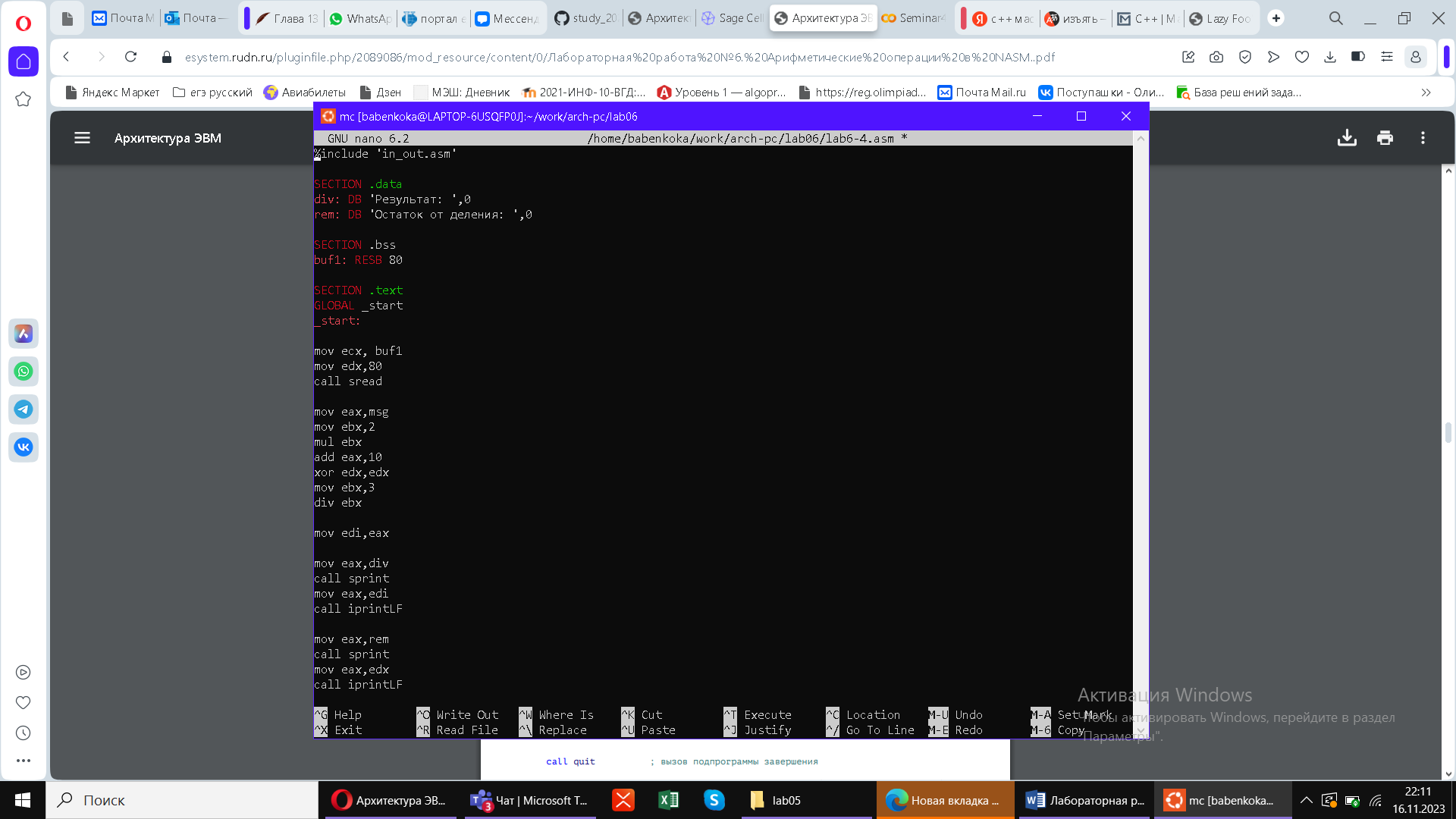
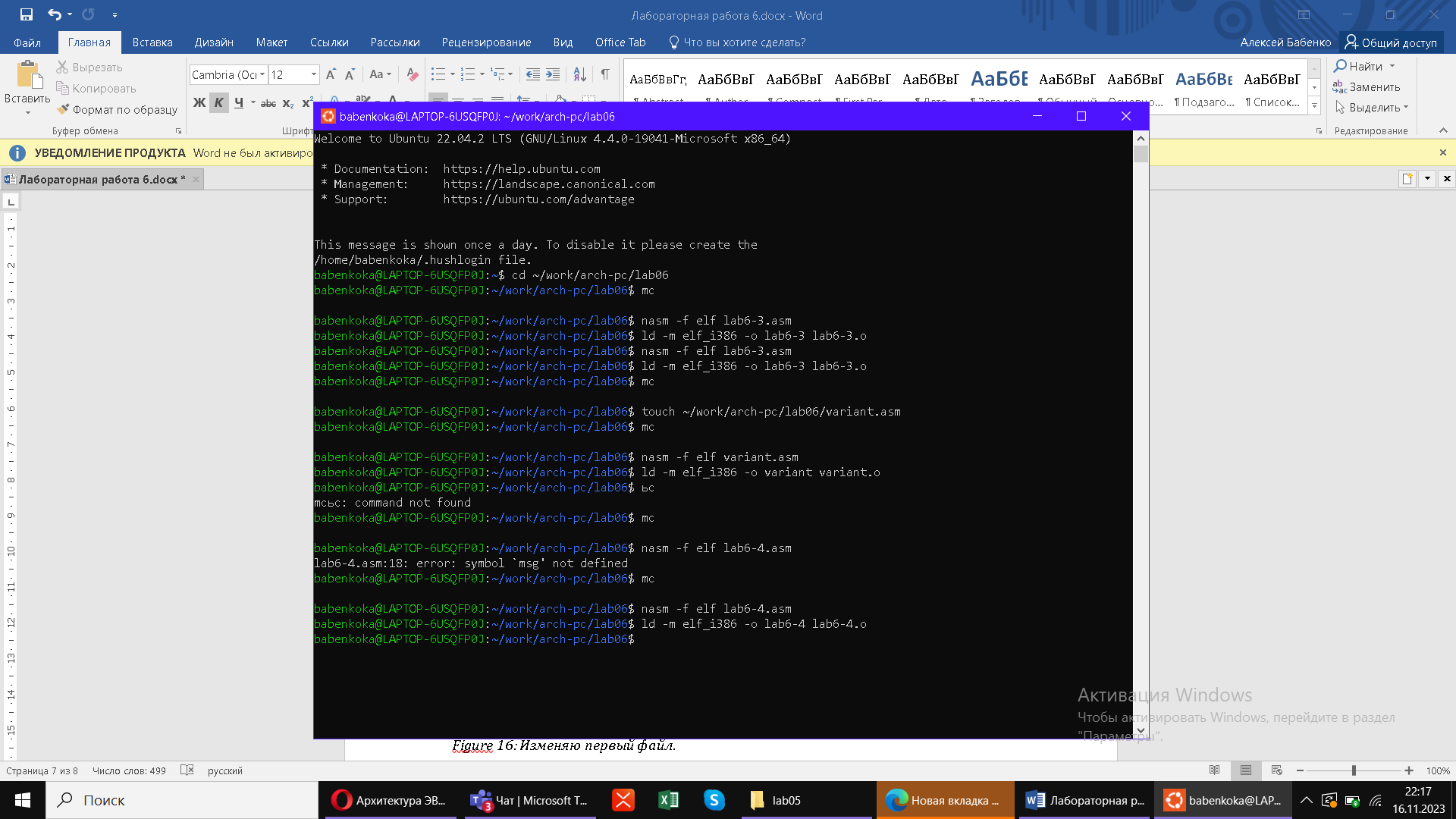


Figure 16: Изменяю первый файл.

Создаю исполняемый файл и проверяю его работу для значений 1 и 10:



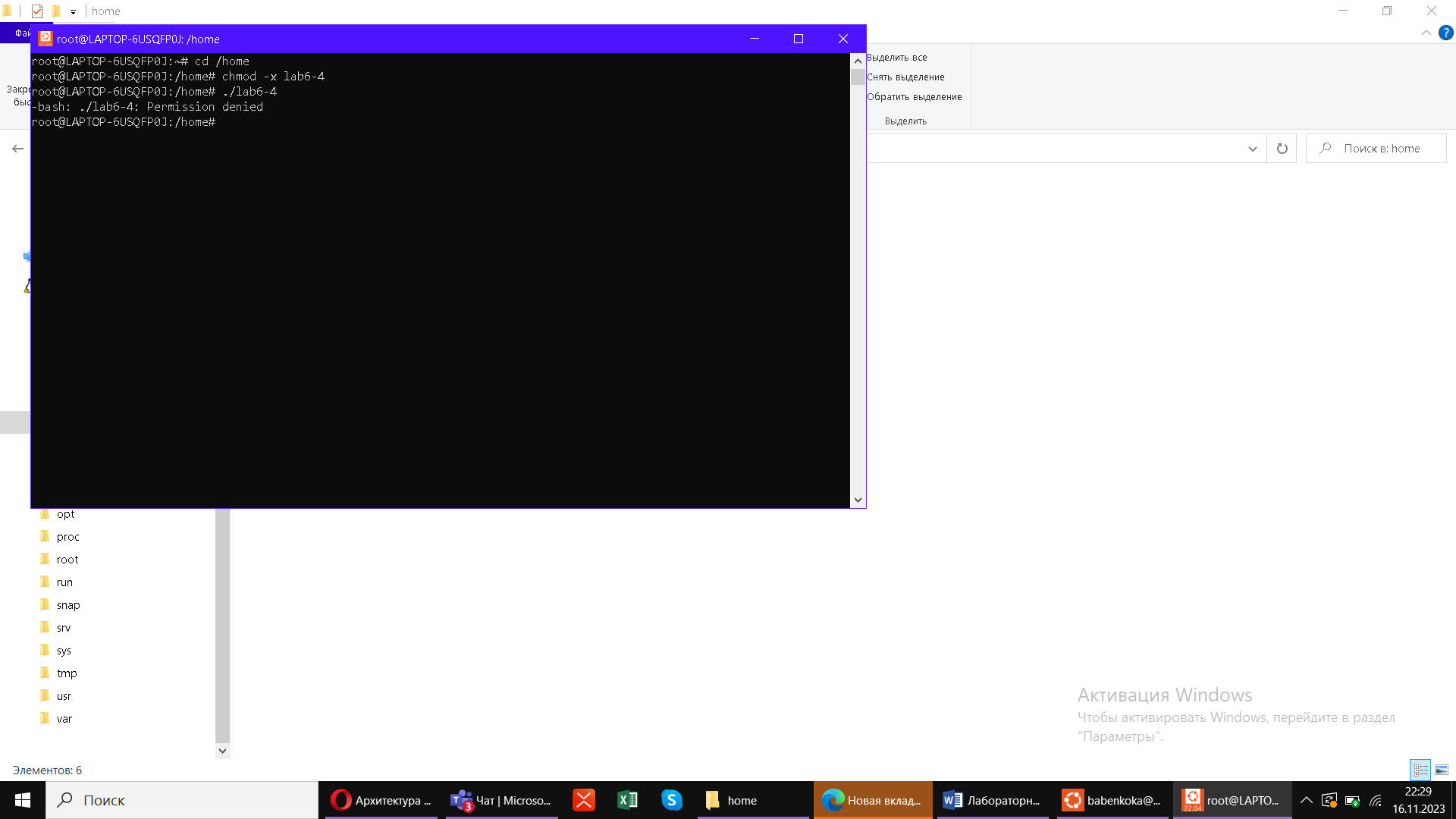


Figure 17: Работа файла для 1.

# 3 Выводы

# Я освоил арифметические инструкции языка ассемблера NASM.