Шаблон отчёта по лабораторной работе

No 07

НВЕ МАНГЕ ХОСЕ ХЕРСОН МИКО, Группа: НКАбд-03-22

Содержание

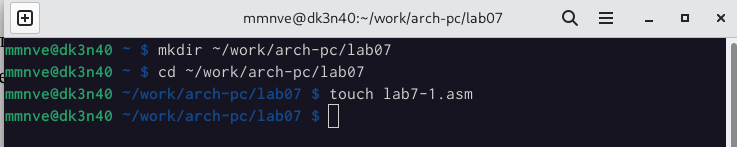
# 1 Цель работы

В седьмой лабораторной работе можно будет освоить арифметические опе- рации языка ассемблера.

# 2 Выполнение лабораторной работы :

## 2.1 Символьные и численные данные в NASM :

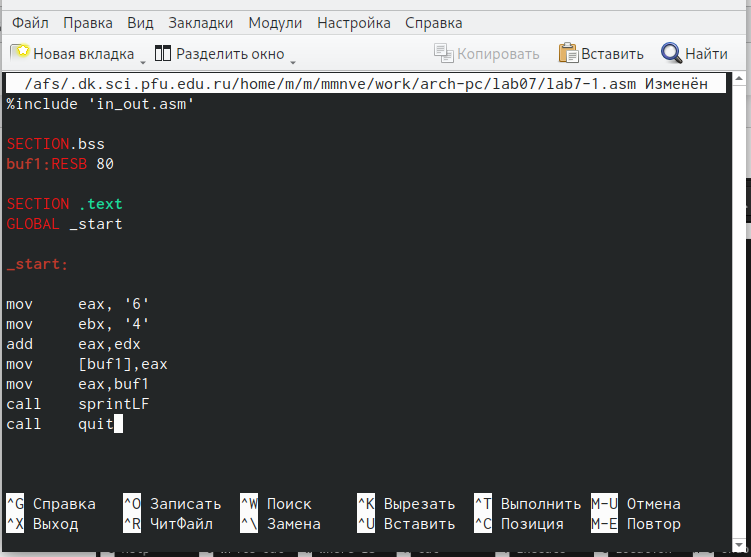
1. Здесь мы начали с создания, а затем переместились в седьмой каталог ла- боратории “~/work/arch-pc/lab07”, после чего мы создали файл “lab7-1.asm”.(рис. [-??)



Ресунек 1

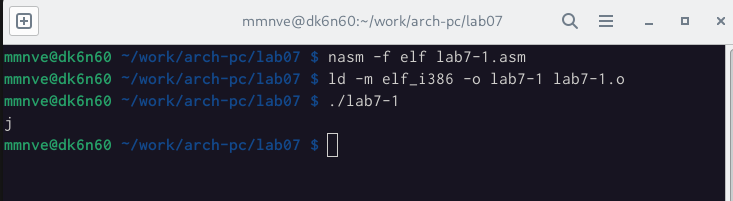
/ /

1. После этого мы заполнили файл .asm кодом программы, отображающей значение регистра eax.(рис. [-??)



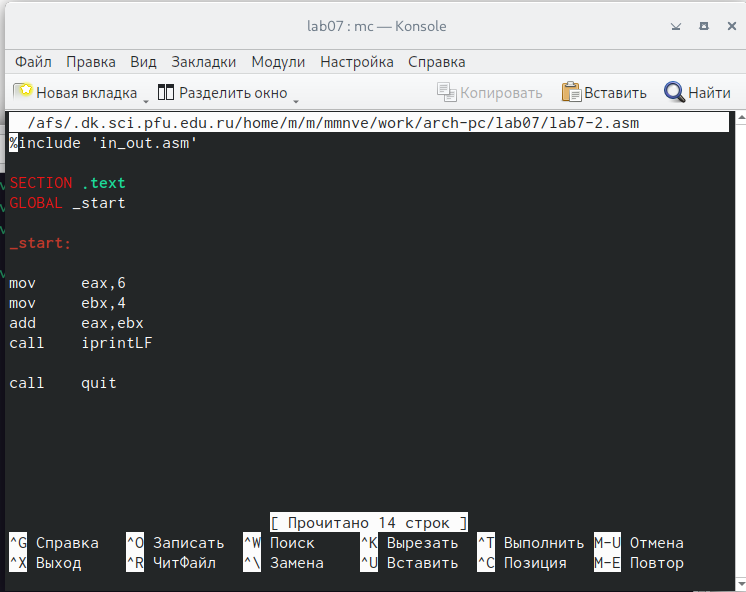
Ресунек 2

* Затем мы скомпилировали файл, создали исполняемый файл и запустили программу, все это после перемещения файла in\_out.asm в тот же каталог, где находится lab7-1.asm.(рис. [-??)



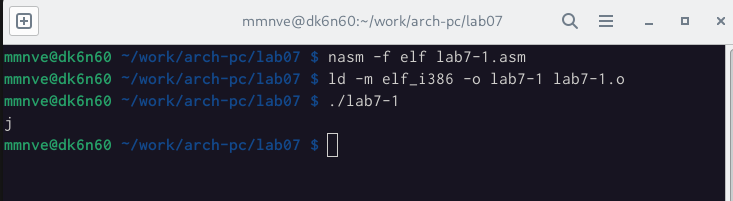
Ресунек 3

1. После этого мы изменили код в листинге следующим образом : mov eax,6 mov ebx,4. (рис. [-??)



Ресунек 4

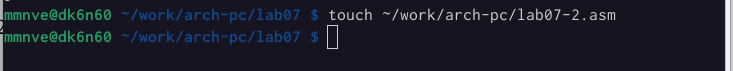
* Затем мы снова скомпилировали файл и создали исполняемый файл.(рис. [-??)



Ресунек 5

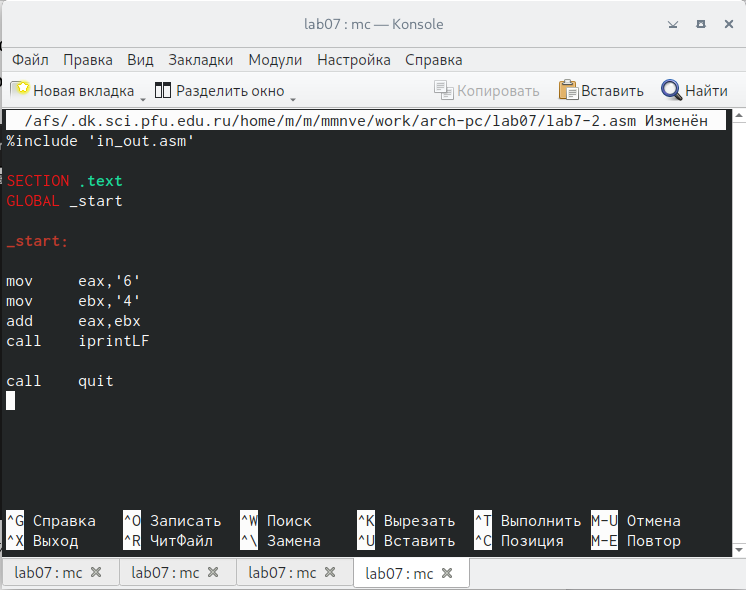
* Проверив ASCII tbale символ, соответствующий коду 10 это новая строка, и мы можем сказать, что это было отображено, потому что при запуске программы она отобразила новую строку в качестве вывода.

1. После этого мы создали файл lab-2.asm, в котором мы использовали под- программы, расположенные в файле in\_out.asm.(рис. [-??)



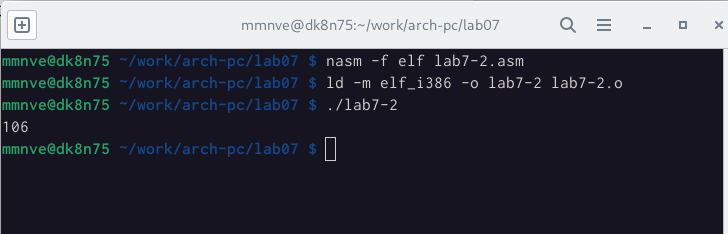
Ресунек 6

* После этого мы заполнили файл необходимым кодом для вывода значения реестра с помощью подпрограммы. (рис. [-??)



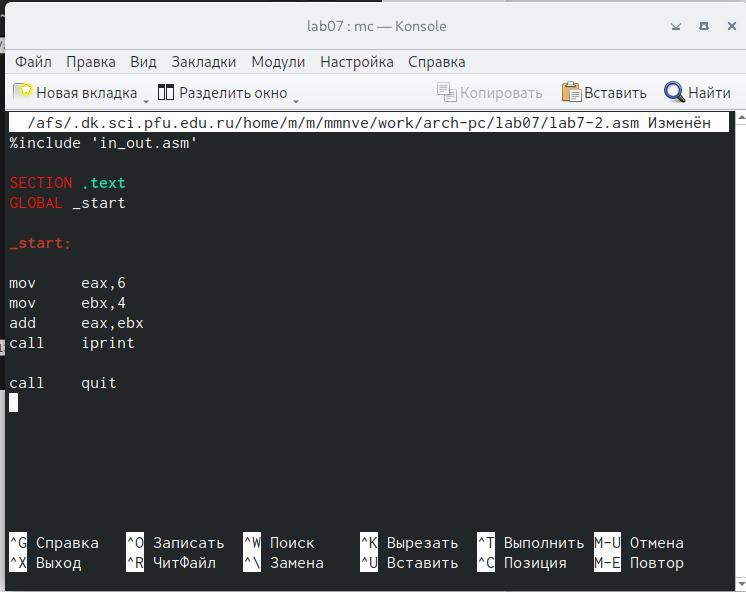
Ресунек 7

* мы скомпилировали файл, создали исполняемый файл и запустили его.(рис. [-??)



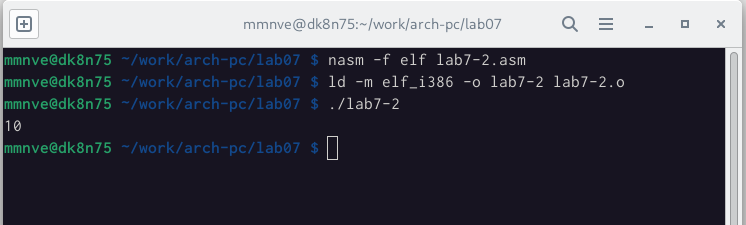
Ресунек 8

1. Аналогично предыдущему примеру, мы меняем символы на цифры, заменяя строки на : mov eax,6 mov ebx,4 (рис. [-??)



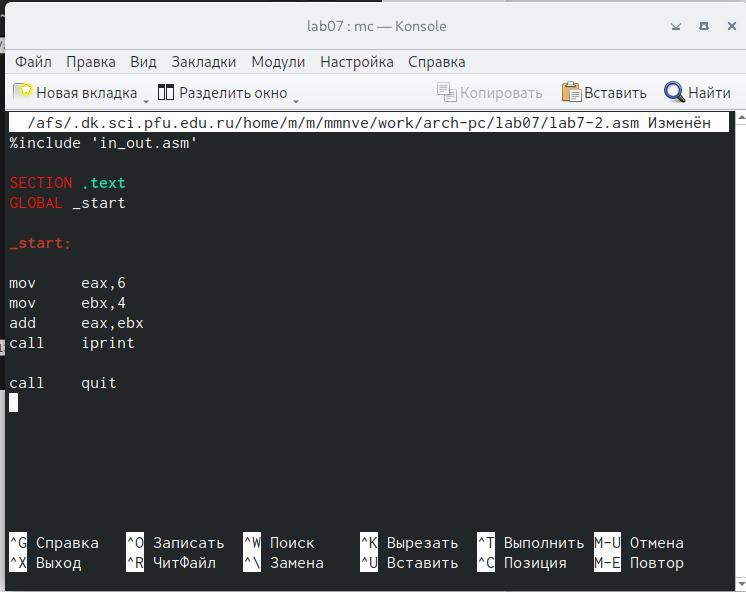
Ресунек 9

* Затем мы снова скомпилировали файл и создали исполняемый файл.(рис. [-??)

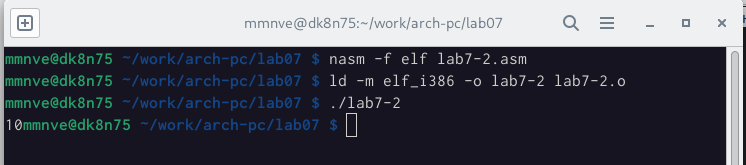


Ресунек 10

* На этот раз результатом, который мы получили, действительно было добав- ление 6 и 4 который 10.
* Затем мы заменили функцию iprintLF на iprint. После этого был создан исполняемый файл, и мы запустили его.(рис.[-??) (рис. [-??)



Ресунек 11



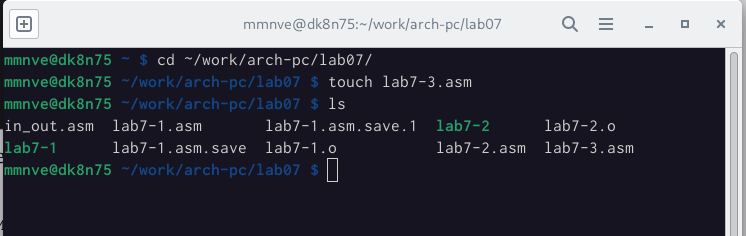
Ресунек 12

* Когда мы использовали подпрограмму iprint, мы заметили, что вывод отли- чается от предыдущего,потому что при использовании iprint не создается новая строка после вывода.

## 2.2 Выполнение арифметических операций в NASM :

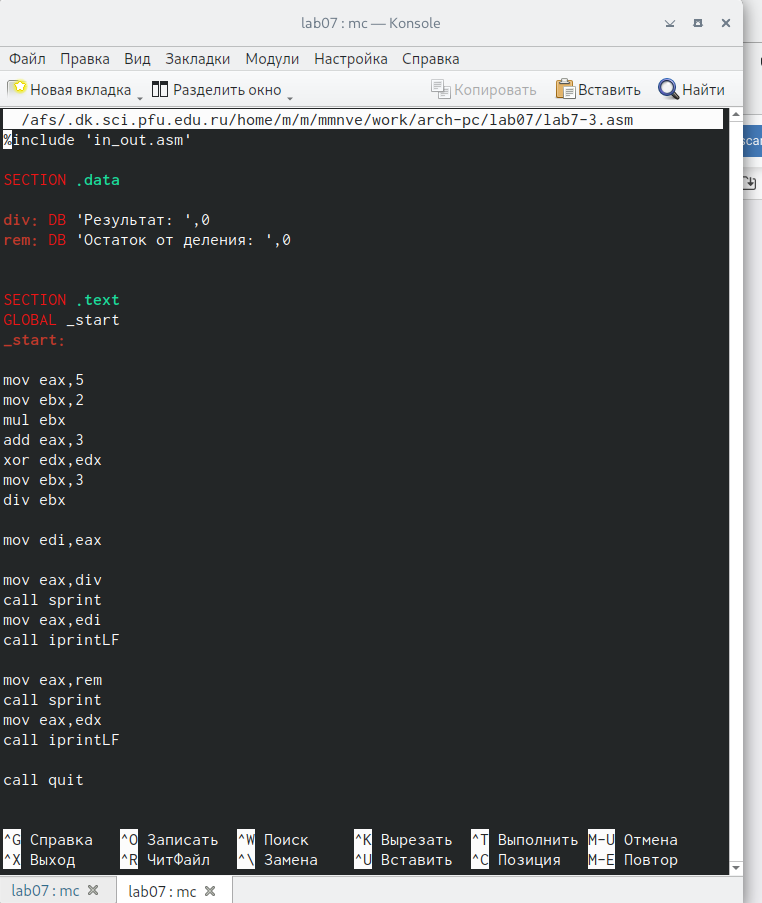
1. В качестве примера выполнения арифметических операций в NASM при- ведем программу вычисления арифметического выражения 𝑓(𝑥) = (5 ∗ 2 + 3)/3

* Мы создали файл lab7-3.asm в каталоге ~/work/arch-pc/lab07.(рис [-??)



Ресунек 13

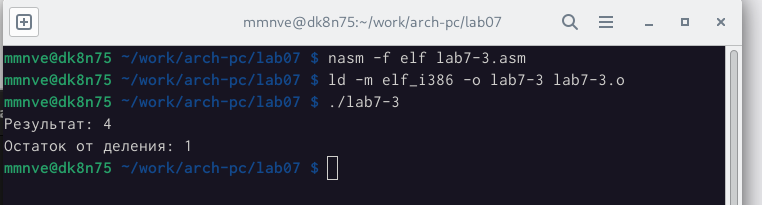
1. Затем мы заполнили файл необходимым кодом.(рис. [-??)



Ресунек 14

* Создали исполняемый файл и запустили его.

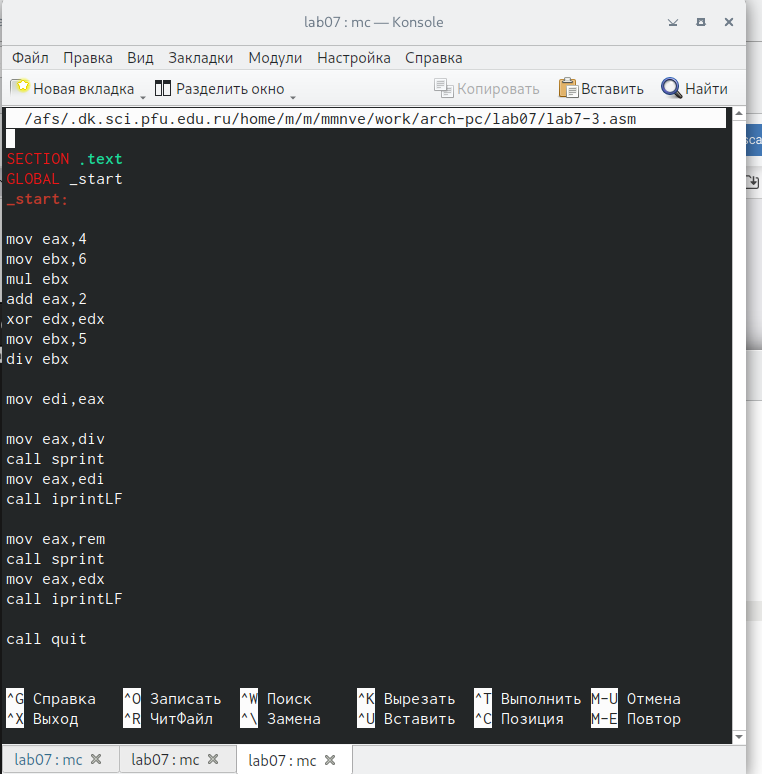
( рис.[-??)



Ресунек 15

* Затем мы изменили текст программы, чтобы вычислить выражение: 𝑓(𝑥) = (4 ∗ 6 + 2)/5.

(рис. [-??)



Ресунек 16

* мы создали исполняемый файл и проверили его работу.(рис. [-??)



Ресунек 17

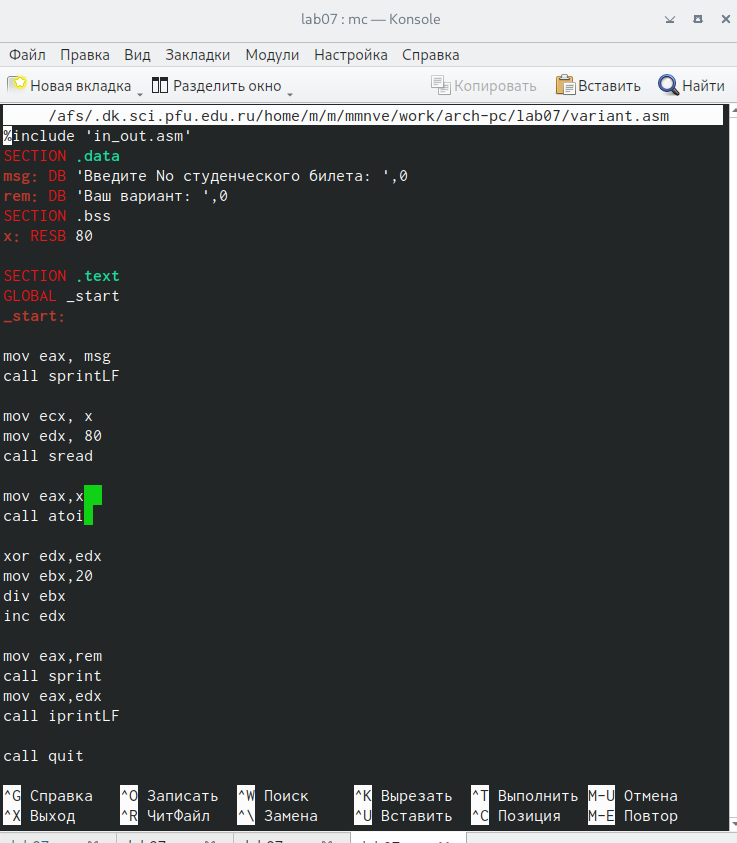
1. На этом шаге мы написали программу, которая может вычислить диспер- сию, которую мы получаем из номера студенческого билета.

* Мы начали с создания файла variant.asm.(рис. [-??)



Ресунек 18

* После этого мы написали код программы. (рис. [-??)



Ресунек 19

* мы создали исполняемый файл и проверили его работу, и действительно, в зависимости от номера студента он генерирует номер варианта. (рис. [-??)



Ресунек 20

## 2.3 Вопросы :

1. Какие строки листинга 7.4 отвечают за вывод на экран сообщения ‘Ваш вариант:’? O: rem: DB ‘Ваш вариант:’,0 mov eax,rem call sprint
2. Для чего используется следующие инструкции? mov ecx, x mov edx, 80 call sread

O: Эти инструкции были использованы для того, чтобы позволить пользовате- лю вводить данные 3. Для чего используется инструкция “call atoi”? O: Эта инструкция используется для преобразования значения x из ASCII-кода в целое число.

1. Какие строки листинга 7.4 отвечают за вычисления варианта? O: xor edx,edx mov ebx,20 div ebx inc edx
2. В какой регистр записывается остаток от деления при выполнении ин- струкции “div ebx”? O:Остаток был записан в регистре edx
3. Для чего используется инструкция “inc edx”? O: Эта инструкция была использована для увеличения значения в регистре edx
4. Какие строки листинга 7.4 отвечают за вывод на экран результата вычис- лений? O: mov eax,edx call iprintLF

## 2.4 Выводы по результатам выполнения заданий :

* В ходе лабораторной работы мы освоили выполнение арифметических опе- раций на языке ассемблера и углубились в использование подпрограммы.

# 3 Задание для самостоятельной работы :

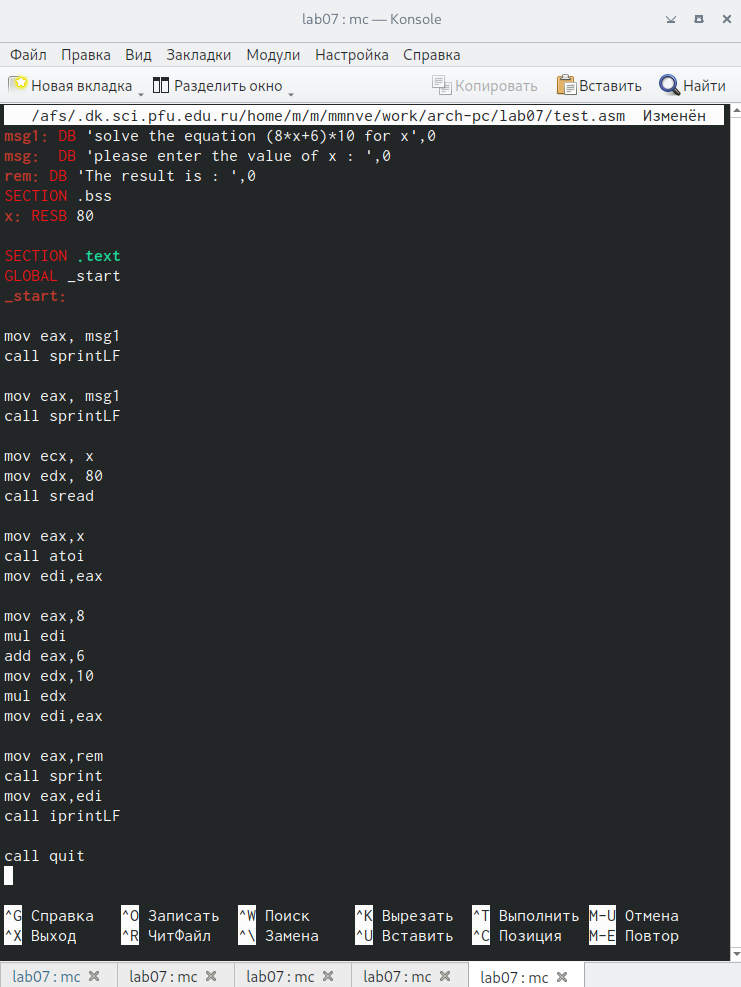
1. В этой работе нам пришлось написать программу, которая просит пользо- вателя ввести значение переменной и решить математическое выражение.

* Мой вариант : 13
* математическое выражение (8𝑥 + 6) ⋅ 10
* Итак, мы начали с создания asm-файла, в котором будет находиться нашкод.(рис. [-??)



Ресунек 21

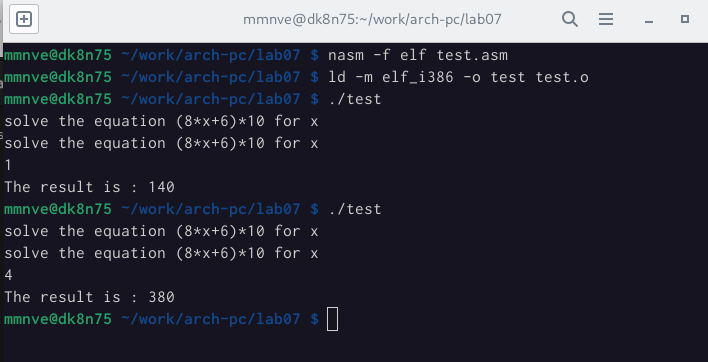
* После этого мы написали код нашей программы. (рис. [-??)



Ресунек 22

* и, наконец, мы проверяем корректность кода, который мы написали, ис- пользуя два разных значения. 𝑥1 = 1 𝑥2 = 4

Как указано на следующем рисунке.(рис. [-??)



Ресунек 23

## 3.1 Выводы по результатам выполнения заданий :

* В этой части мы смогли узнать, как преобразовать некоторые математические идеи в реальный код на ассемблере, что помогло нам получить более глубокое представление о том, как работать с регистрами.

# 4 Выводы, согласованные с целью работы :

* В седьмой лаборатории мы в основном научились писать программы, вы- полняющие арифметические операции, и научились вычислять математи- ческие выражения средней сложности.