

Шаблон отчёта по лабораторной работе

№ 07

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

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы :	6
2.1	Символьные и численные данные в NASM :	6
2.2	Выполнение арифметических операций в NASM :	16
2.3	Вопросы :	22
2.4	Выводы по результатам выполнения заданий :	23
3	Задание для самостоятельной работы :	24
3.1	Выводы по результатам выполнения заданий :	26
4	Выводы, согласованные с целью работы :	27

Список иллюстраций

2.1	Ресунок 1	6
2.2	Ресунок 2	7
2.3	Ресунок 3	8
2.4	Ресунок 4	9
2.5	Ресунок 5	10
2.6	Ресунок 6	10
2.7	Ресунок 7	11
2.8	Ресунок 8	12
2.9	Ресунок 9	13
2.10	Ресунок 10	14
2.11	Ресунок 11	15
2.12	Ресунок 12	15
2.13	Ресунок 13	16
2.14	Ресунок 14	17
2.15	Ресунок 15	18
2.16	Ресунок 16	19
2.17	Ресунок 17	20
2.18	Ресунок 18	20
2.19	Ресунок 19	21
2.20	Ресунок 20	22
3.1	Ресунок 21	24
3.2	Ресунок 22	25
3.3	Ресунок 23	26

Список таблиц

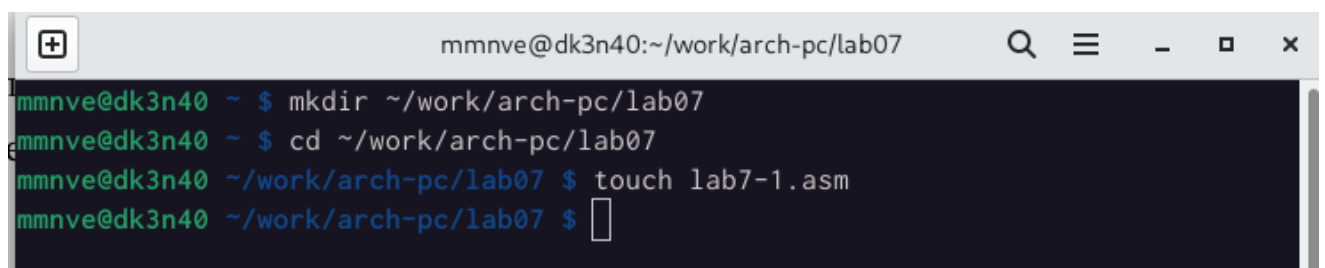
1 Цель работы

В седьмой лабораторной работе можно будет освоить арифметические операции 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 лаборатории “~/work/arch-pc/lab07”, после чего мы создали файл “lab7-1.asm”.(рис. [-2.1])

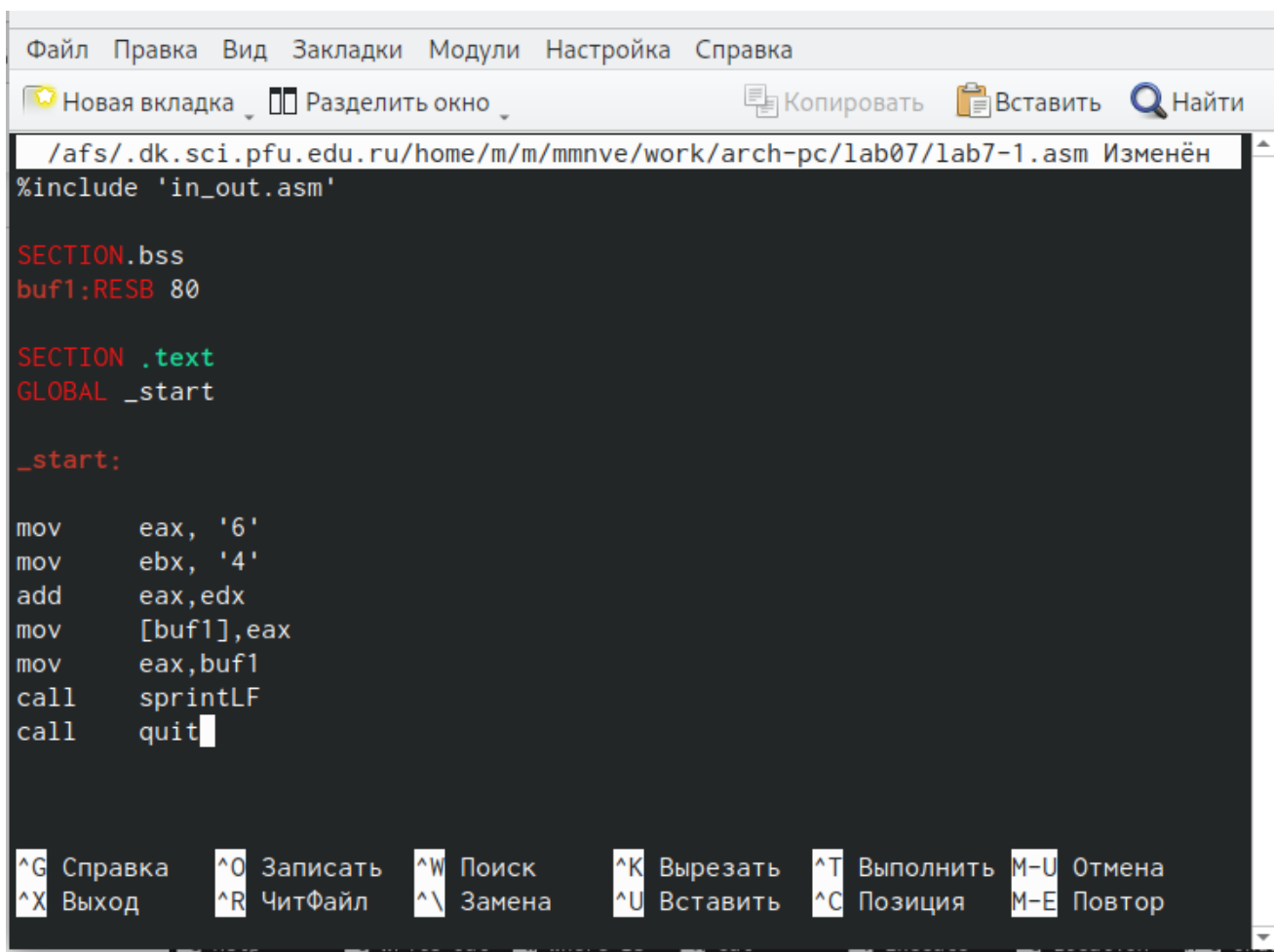


```
mmnve@dk3n40:~/work/arch-pc/lab07
mmnve@dk3n40 ~ $ mkdir ~/work/arch-pc/lab07
mmnve@dk3n40 ~ $ cd ~/work/arch-pc/lab07
mmnve@dk3n40 ~/work/arch-pc/lab07 $ touch lab7-1.asm
mmnve@dk3n40 ~/work/arch-pc/lab07 $
```

Рис. 2.1: Ресунок 1

//

2. После этого мы заполнили файл .asm кодом программы, отображающей значение регистра eax.(рис. [-2.2])



```
Файл  Правка  Вид  Закладки  Модули  Настройка  Справка
Новая вкладка  Разделить окно  Копировать  Вставить  Найти
/afs/.dk.sci.pfu.edu.ru/home/m/m/mmnve/work/arch-pc/lab07/lab7-1.asm  Изменён
#include 'in_out.asm'

SECTION.bss
buf1:RESB 80

SECTION .text
GLOBAL _start

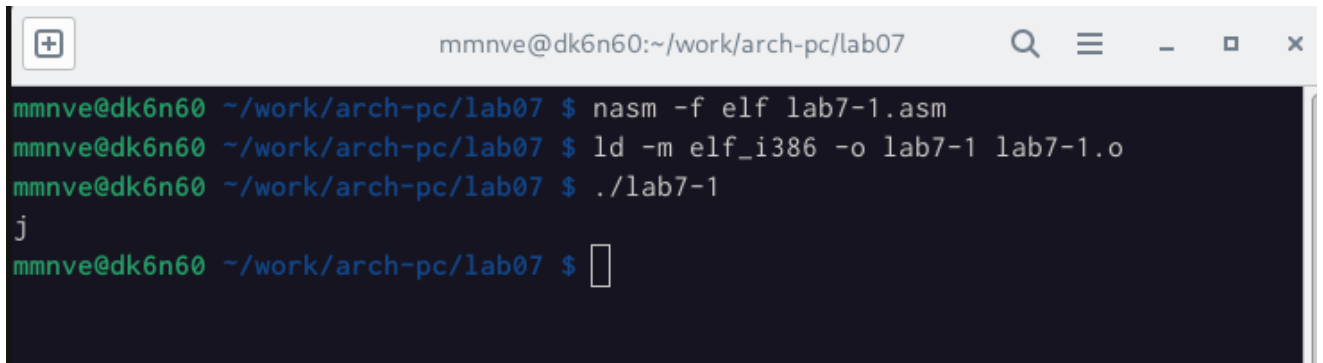
_start:

mov     eax, '6'
mov     ebx, '4'
add     eax,edx
mov     [buf1],eax
mov     eax,buf1
call    sprintf
call    quit
```

^G Справка ^O Записать ^W Поиск ^K Вырезать ^T Выполнить M-U Отмена
^X Выход ^R ЧитФайл ^\ Замена ^U Вставить ^C Позиция M-E Повтор

Рис. 2.2: Ресунок 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. [-2.3)

A terminal window with a dark background and light-colored text. The window title bar shows the user 'mmnve' at host 'dk6n60' in the directory '~/work/arch-pc/lab07'. The terminal contains the following text:

```
mmnve@dk6n60 ~/work/arch-pc/lab07 $ nasm -f elf lab7-1.asm
mmnve@dk6n60 ~/work/arch-pc/lab07 $ ld -m elf_i386 -o lab7-1 lab7-1.o
mmnve@dk6n60 ~/work/arch-pc/lab07 $ ./lab7-1
j
mmnve@dk6n60 ~/work/arch-pc/lab07 $
```

Рис. 2.3: Ресунек 3

3. После этого мы изменили код в листинге следующим образом : `mov eax,6`
`mov ebx,4.` (рис. [-2.4)


```
lab07: mc — Konsole
Файл  Правка  Вид  Закладки  Модули  Настройка  Справка
Новая вкладка  Разделить окно  Копировать  Вставить  Найти
/afs/.dk.sci.pfu.edu.ru/home/m/m/mmnve/work/arch-pc/lab07/lab7-2.asm
%include 'in_out.asm'

SECTION .text
GLOBAL _start

_start:

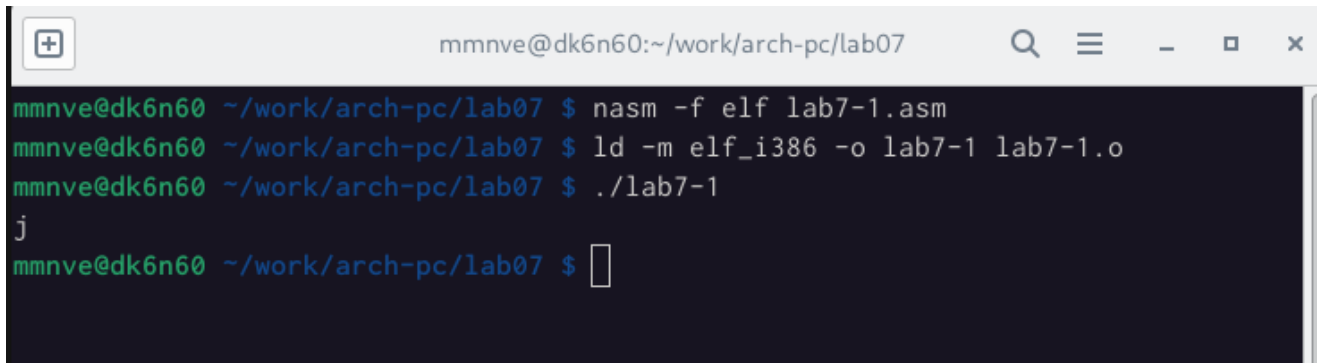
mov     eax,6
mov     ebx,4
add     eax,ebx
call    iprintLF

call    quit

[ Прочитано 14 строк ]
^G Справка  ^O Записать  ^W Поиск  ^K Вырезать  ^T Выполнить  M-U Отмена
^X Выход    ^R ЧитФайл  ^N Замена  ^U Вставить  ^C Позиция  M-E Повтор
```

Рис. 2.4: Ресунок 4

- Затем мы снова скомпилировали файл и создали исполняемый файл.(рис. [-2.5)

A terminal window with a dark background and light green text. The title bar shows the user 'mmnve' at host 'dk6n60' in the directory '~/work/arch-pc/lab07'. The terminal contains the following commands and output:

```
mmnve@dk6n60 ~/work/arch-pc/lab07 $ nasm -f elf lab7-1.asm
mmnve@dk6n60 ~/work/arch-pc/lab07 $ ld -m elf_i386 -o lab7-1 lab7-1.o
mmnve@dk6n60 ~/work/arch-pc/lab07 $ ./lab7-1
j
mmnve@dk6n60 ~/work/arch-pc/lab07 $
```

Рис. 2.5: Ресунок 5

- Проверив ASCII table 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.
4. После этого мы создали файл lab-2.asm, в котором мы использовали подпрограммы, расположенные в файле in_out.asm.(рис. [-2.6)

A terminal window with a dark background and light green text. The title bar shows the user 'mmnve' at host 'dk6n60' in the directory '~/work/arch-pc/lab07'. The terminal contains the following commands and output:

```
mmnve@dk6n60 ~/work/arch-pc/lab07 $ touch ~/work/arch-pc/lab07-2.asm
mmnve@dk6n60 ~/work/arch-pc/lab07 $
```

Рис. 2.6: Ресунок 6

- После этого мы заполнили файл необходимым кодом для вывода значения реестра с помощью подпрограммы. (рис. [-2.7)

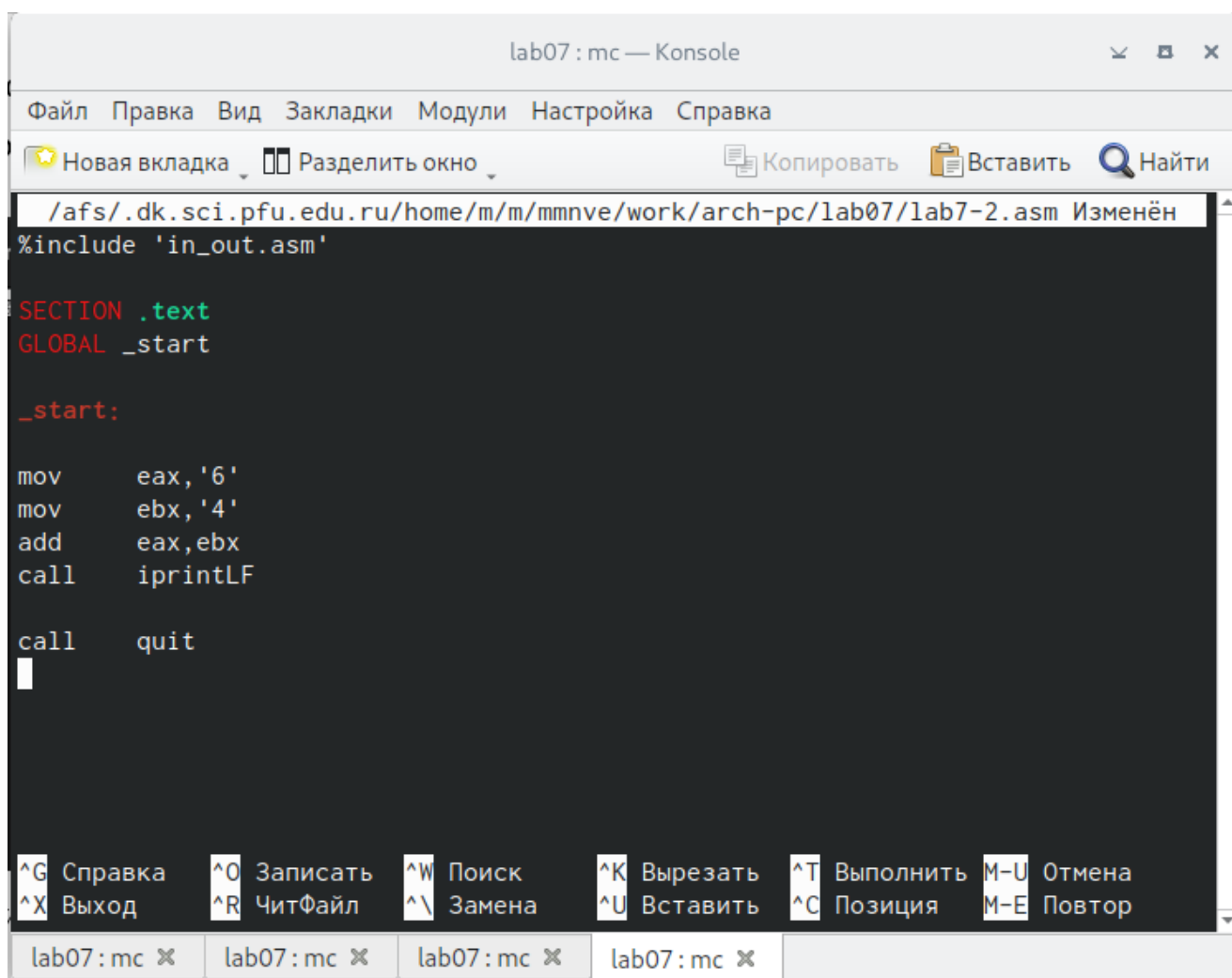
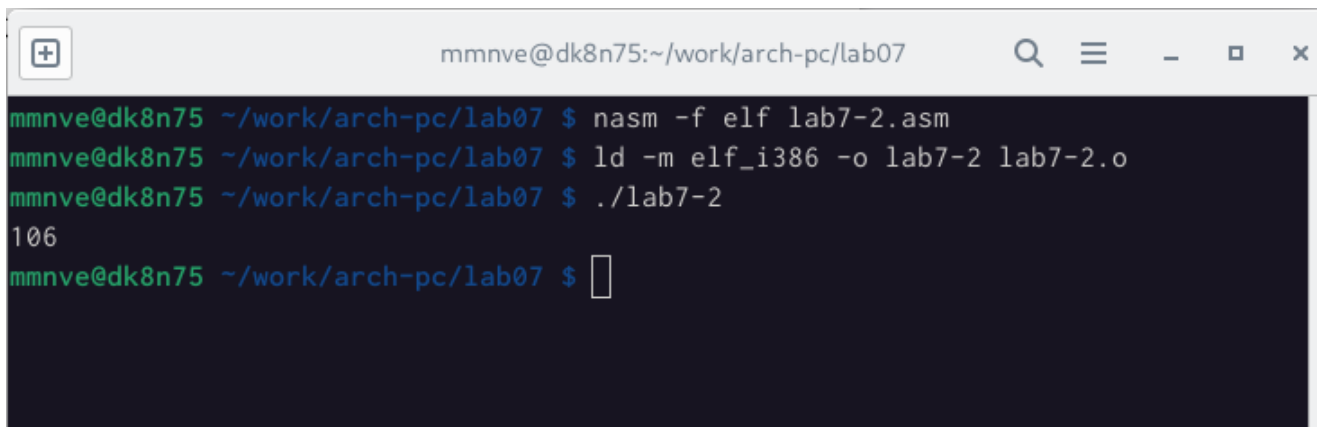


Рис. 2.7: Ресунок 7

- мы скомпилировали файл, создали исполняемый файл и запустили его.(рис. [-2.8)

A terminal window with a title bar showing the user 'mmnve' at host 'dk8n75' in the directory '~/work/arch-pc/lab07'. The terminal contains the following commands and output:

```
mmnve@dk8n75 ~/work/arch-pc/lab07 $ nasm -f elf lab7-2.asm
mmnve@dk8n75 ~/work/arch-pc/lab07 $ ld -m elf_i386 -o lab7-2 lab7-2.o
mmnve@dk8n75 ~/work/arch-pc/lab07 $ ./lab7-2
106
mmnve@dk8n75 ~/work/arch-pc/lab07 $
```

Рис. 2.8: Ресунок 8

5. Аналогично предыдущему примеру, мы меняем символы на цифры, заменяя строки на : `mov eax,6` `mov ebx,4` (рис. [-2.9])

```
lab07: mc — Konsole
Файл  Правка  Вид  Закладки  Модули  Настройка  Справка
Новая вкладка  Разделить окно  Копировать  Вставить  Найти
/afs/.dk.sci.pfu.edu.ru/home/m/m/mmnve/work/arch-pc/lab07/lab7-2.asm Изменён
%include 'in_out.asm'

SECTION .text
GLOBAL _start

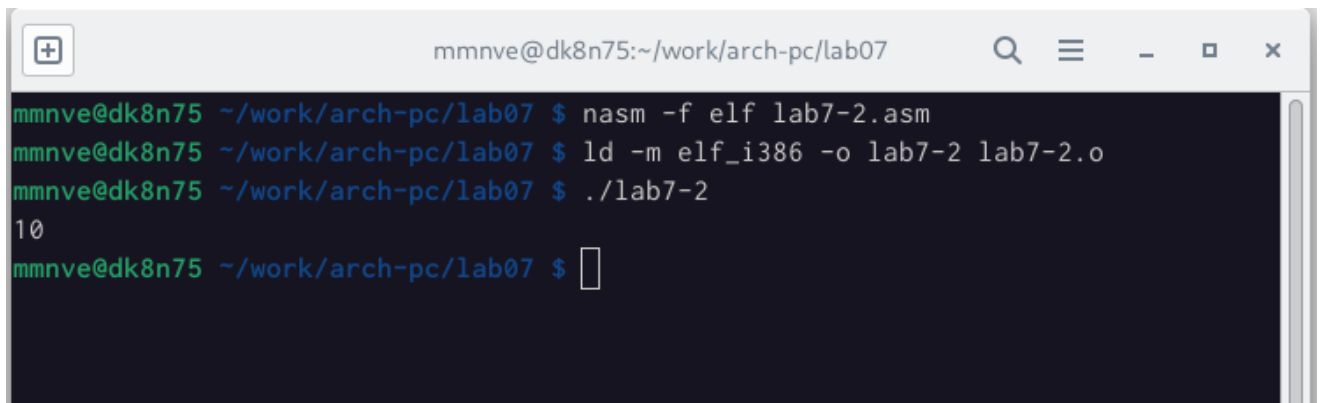
_start:

mov     eax, 6
mov     ebx, 4
add     eax, ebx
call    iprint
call    quit
```

^G Справка ^O Записать ^W Поиск ^K Вырезать ^T Выполнить M-U Отмена
^X Выход ^R ЧитФайл ^\ Замена ^U Вставить ^C Позиция M-E Повтор

Рис. 2.9: Ресунек 9

- Затем мы снова скомпилировали файл и создали исполняемый файл.(рис. [-2.10)

A terminal window with a title bar showing the user 'mmnve' at host 'dk8n75' in the directory '~/work/arch-pc/lab07'. The terminal contains the following commands and output:

```
mmnve@dk8n75 ~/work/arch-pc/lab07 $ nasm -f elf lab7-2.asm
mmnve@dk8n75 ~/work/arch-pc/lab07 $ ld -m elf_i386 -o lab7-2 lab7-2.o
mmnve@dk8n75 ~/work/arch-pc/lab07 $ ./lab7-2
10
mmnve@dk8n75 ~/work/arch-pc/lab07 $
```

Рис. 2.10: Ресунок 10

- На этот раз результатом, который мы получили, действительно было добавление 6 и 4 который 10.
- Затем мы заменили функцию `iprintLF` на `iprint`. После этого был создан исполняемый файл, и мы запустили его.(рис.[2.11) (рис. [2.12)

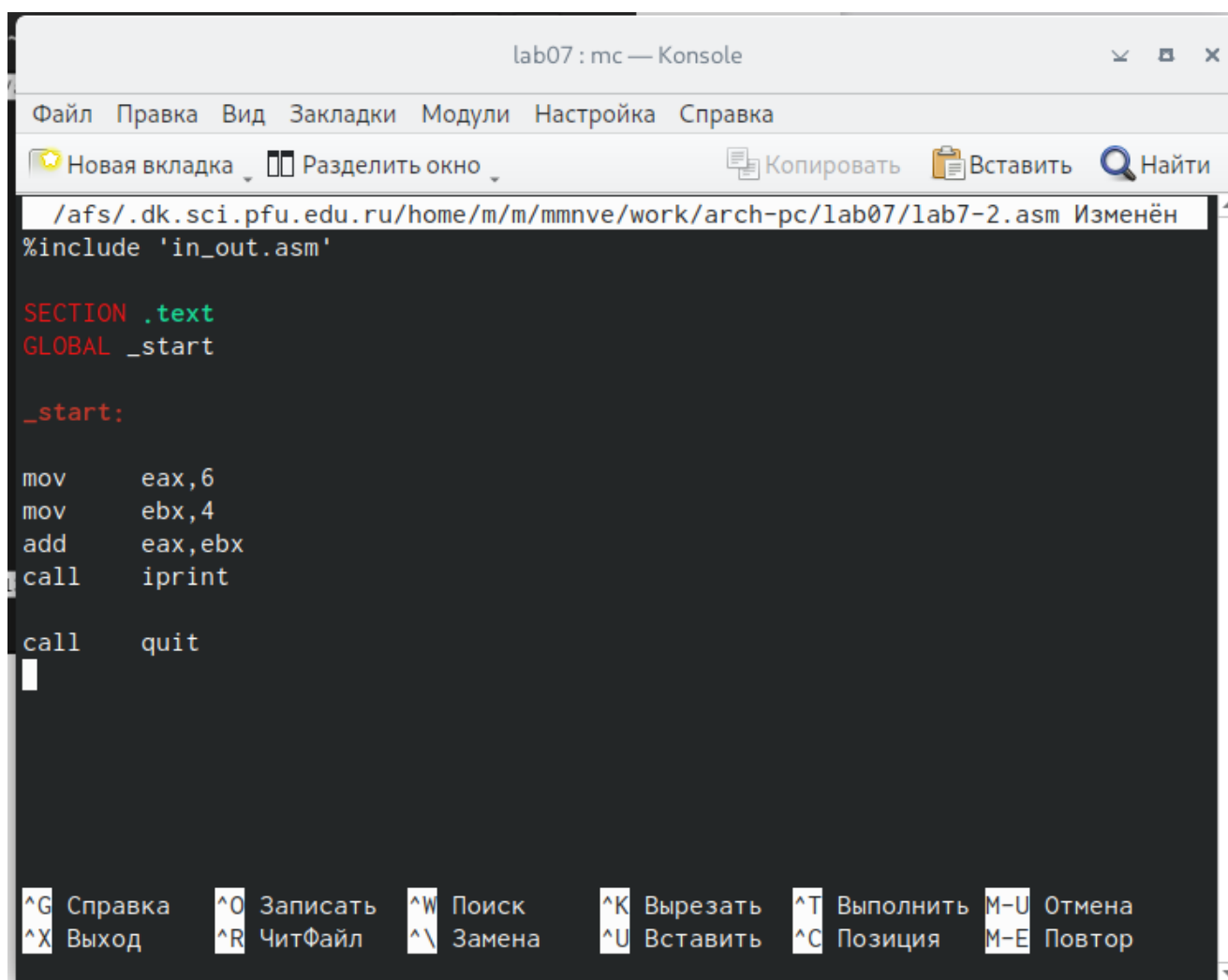


Рис. 2.11: Ресунок 11



Рис. 2.12: Ресунок 12

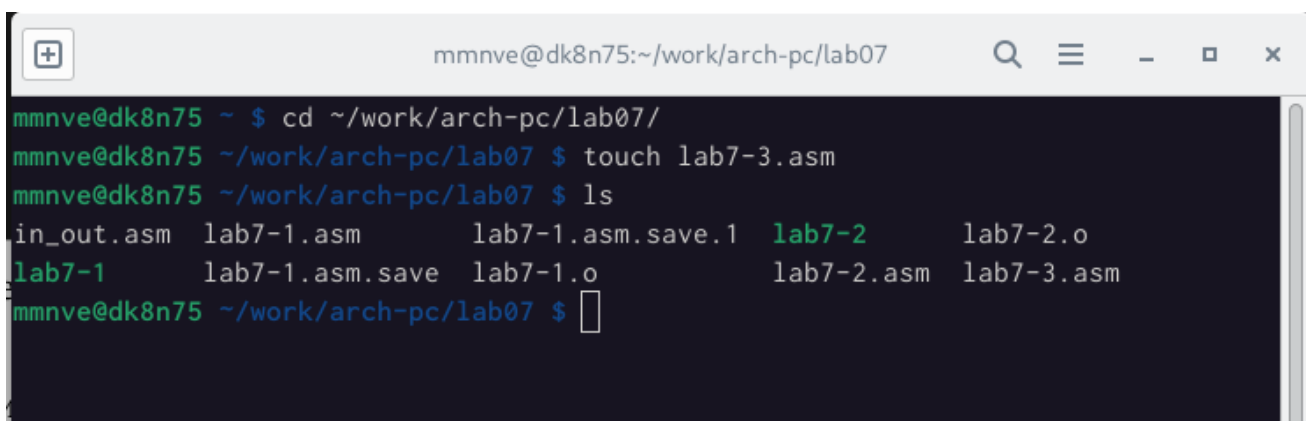
- Когда мы использовали подпрограмму `iprint`, мы заметили, что вывод отли-

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 приведем программу вычисления арифметического выражения $\boxed{x}(\boxed{x}) = (5 \boxed{x} 2 + 3)/3$

 - Мы создали файл `lab7-3.asm` в каталоге `~/work/arch-pc/lab07`. (рис [-2.13])



```
mmnve@dk8n75:~/work/arch-pc/lab07
mmnve@dk8n75 ~ $ cd ~/work/arch-pc/lab07/
mmnve@dk8n75 ~/work/arch-pc/lab07 $ touch lab7-3.asm
mmnve@dk8n75 ~/work/arch-pc/lab07 $ ls
in_out.asm  lab7-1.asm      lab7-1.asm.save.1  lab7-2      lab7-2.o
lab7-1      lab7-1.asm.save lab7-1.o           lab7-2.asm  lab7-3.asm
mmnve@dk8n75 ~/work/arch-pc/lab07 $
```

Рис. 2.13: Ресунок 13

2. Затем мы заполнили файл необходимым кодом. (рис. [-2.14])

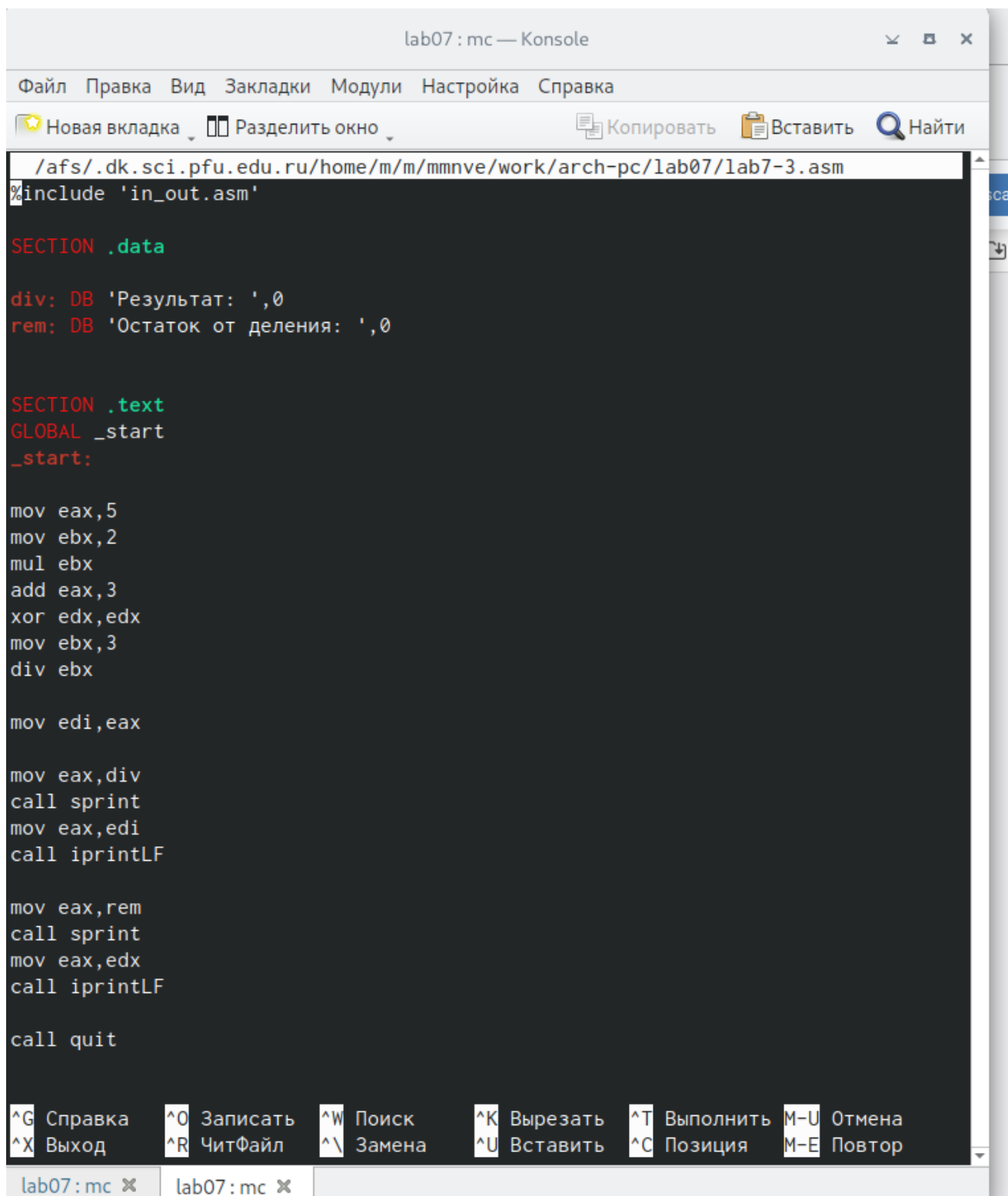
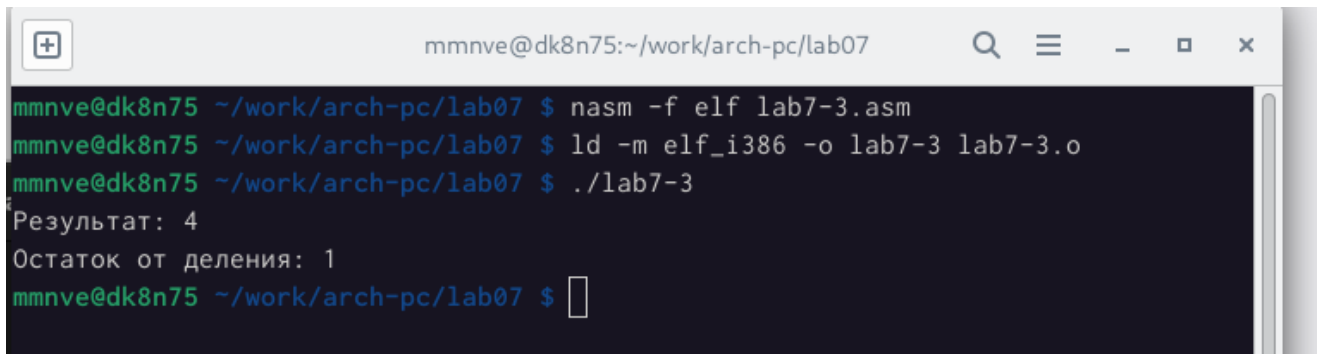


Рис. 2.14: Ресунек 14

- Создали исполняемый файл и запустили его.

(рис.[-2.15)



```
mmnve@dk8n75:~/work/arch-pc/lab07
mmnve@dk8n75 ~/work/arch-pc/lab07 $ nasm -f elf lab7-3.asm
mmnve@dk8n75 ~/work/arch-pc/lab07 $ ld -m elf_i386 -o lab7-3 lab7-3.o
mmnve@dk8n75 ~/work/arch-pc/lab07 $ ./lab7-3
Результат: 4
Остаток от деления: 1
mmnve@dk8n75 ~/work/arch-pc/lab07 $
```

Рис. 2.15: Ресунек 15

- Затем мы изменили текст программы, чтобы вычислить выражение: $\boxtimes(\boxtimes) = (4 \boxtimes 6 + 2)/5$.

(рис. [-2.16)

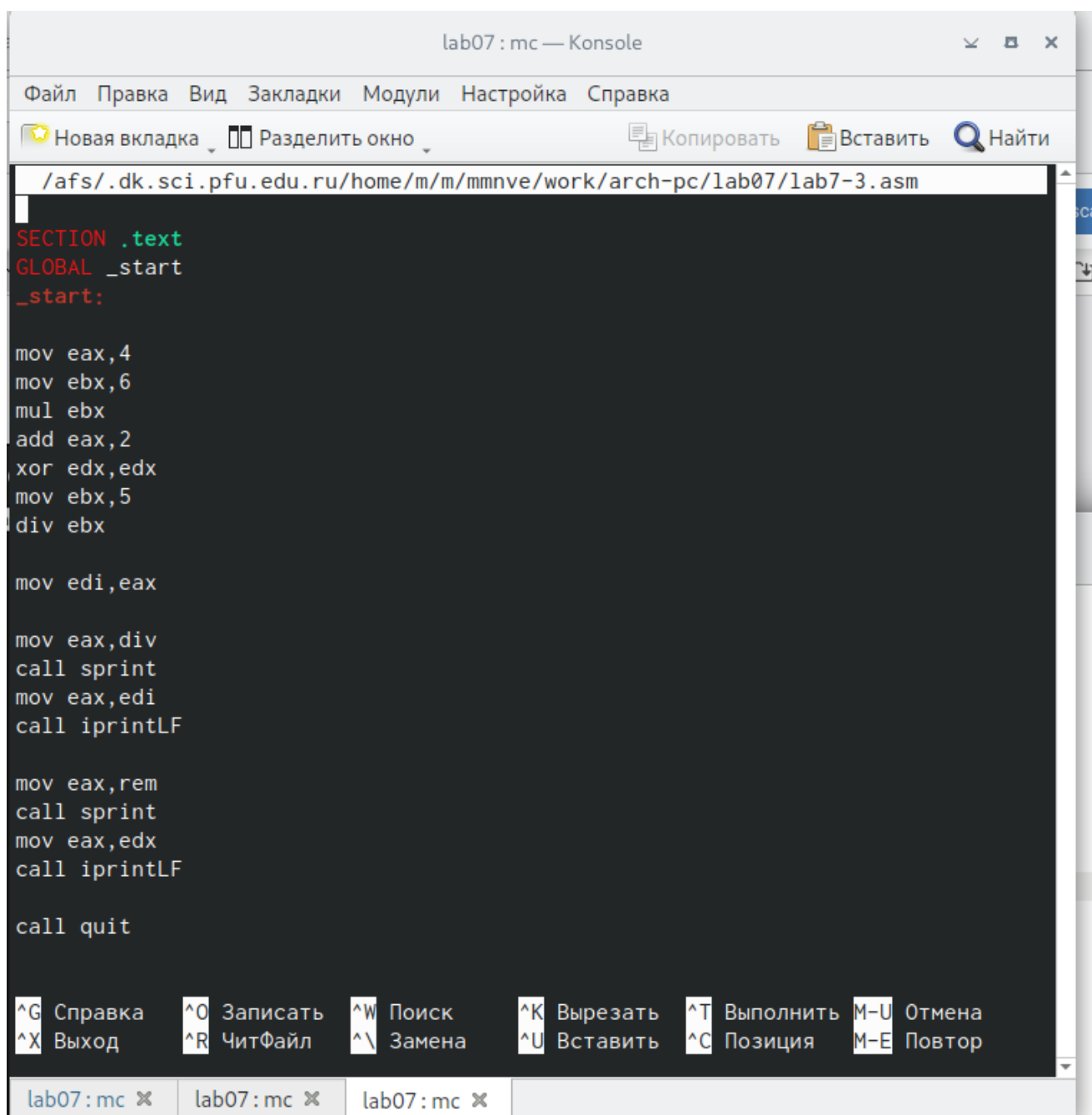


Рис. 2.16: Ресунок 16

- мы создали исполняемый файл и проверили его работу.(рис. [-2.17)



```
mmnve@dk8n75:~/work/arch-pc/lab07
mmnve@dk8n75 ~/work/arch-pc/lab07 $ nasm -f elf lab7-3.asm
mmnve@dk8n75 ~/work/arch-pc/lab07 $ ld -m elf_i386 -o lab7-3 lab7-3.o
mmnve@dk8n75 ~/work/arch-pc/lab07 $ ./lab7-3
Результат: 5
Остаток от деления: 1
mmnve@dk8n75 ~/work/arch-pc/lab07 $
```

Рис. 2.17: Ресунок 17

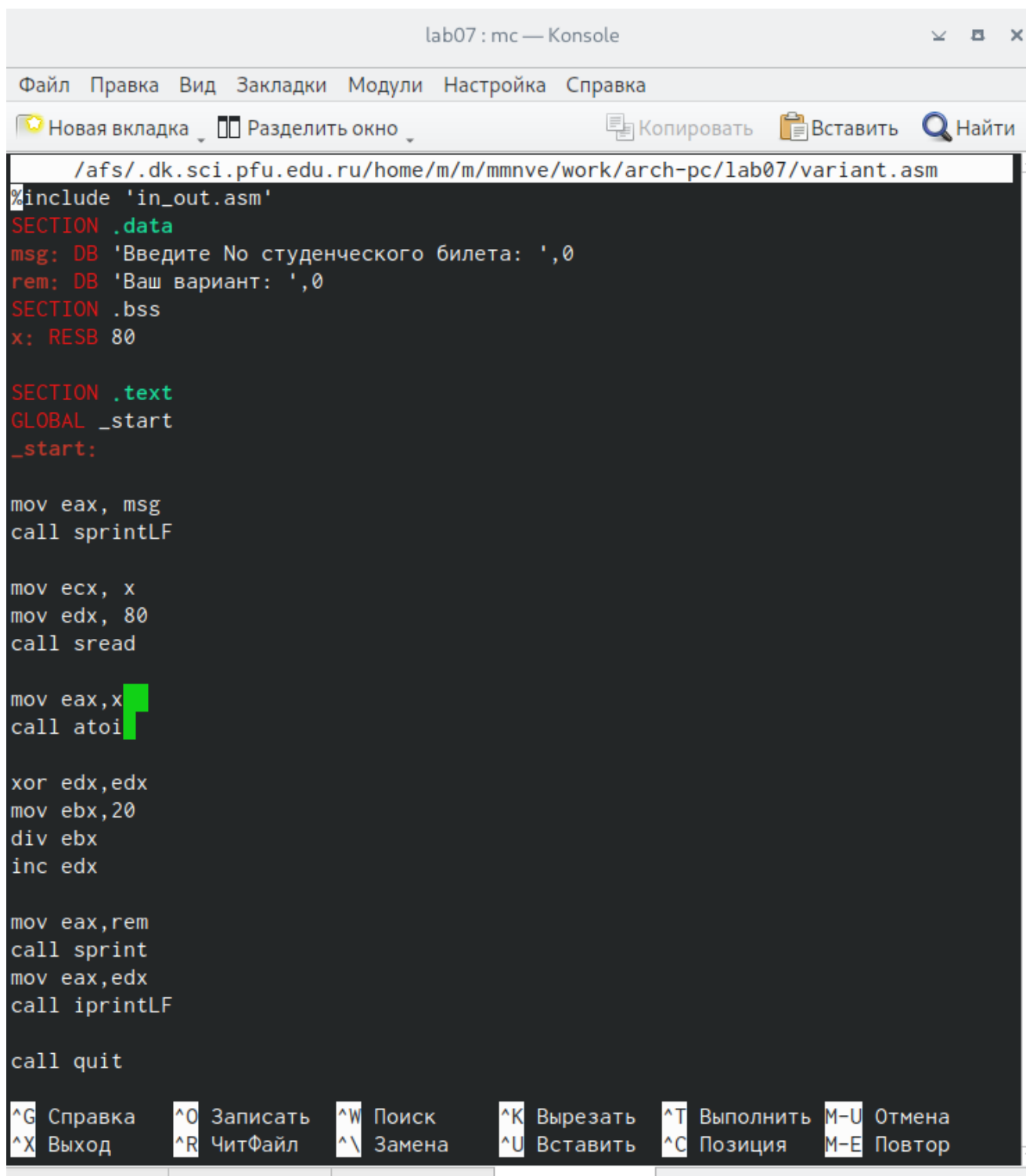
3. На этом шаге мы написали программу, которая может вычислить дисперсию, которую мы получаем из номера студенческого билета.
- Мы начали с создания файла variant.asm.(рис. [-2.18)



```
mmnve@dk8n75:~/work/arch-pc/lab07
mmnve@dk8n75 ~/work/arch-pc/lab07 $ touch variant.asm
mmnve@dk8n75 ~/work/arch-pc/lab07 $ ls
in_out.asm  lab7-1.asm.save  lab7-2      lab7-3      variant.asm
lab7-1      lab7-1.asm.save.1 lab7-2.asm  lab7-3.asm
lab7-1.asm  lab7-1.o         lab7-2.o   lab7-3.o
mmnve@dk8n75 ~/work/arch-pc/lab07 $
```

Рис. 2.18: Ресунок 18

- После этого мы написали код программы. (рис. [-2.19)



The image shows a Konsole window titled "lab07 : mc — Konsole". The window has a menu bar with "Файл", "Правка", "Вид", "Закладки", "Модули", "Настройка", and "Справка". Below the menu bar is a toolbar with icons for "Новая вкладка", "Разделить окно", "Копировать", "Вставить", and "Найти". The address bar shows the file path: "/afs/.dk.sci.pfu.edu.ru/home/m/m/mmnve/work/arch-pc/lab07/variant.asm". The main text area contains the following assembly code:

```
%include 'in_out.asm'
SECTION .data
msg: DB 'Введите No студенческого билета: ',0
rem: DB 'Ваш вариант: ',0
SECTION .bss
x: RESB 80

SECTION .text
GLOBAL _start
_start:

mov eax, msg
call sprintf

mov ecx, x
mov edx, 80
call sread

mov eax, x
call atoi

xor edx, edx
mov ebx, 20
div ebx
inc edx

mov eax, rem
call sprintf
mov eax, edx
call iprintLF

call quit
```

At the bottom of the window, there is a status bar with keyboard shortcuts: ^G Справка, ^O Записать, ^W Поиск, ^K Вырезать, ^T Выполнить, M-U Отмена, ^X Выход, ^R ЧитФайл, ^\ Замена, ^U Вставить, ^C Позиция, M-E Повтор.

Рис. 2.19: Ресунок 19

- мы создали исполняемый файл и проверили его работу, и действительно, в

зависимости от номера студента он генерирует номер варианта. (рис. [-2.20)



```
mmnve@dk8n75:~/work/arch-pc/lab07 $ nasm -f elf variant.asm
mmnve@dk8n75:~/work/arch-pc/lab07 $ ld -m elf_i386 -o variant variant.o
mmnve@dk8n75:~/work/arch-pc/lab07 $ ./variant
Введите No студенческого билета:
1032225355
Ваш вариант: 16
mmnve@dk8n75:~/work/arch-pc/lab07 $
```

Рис. 2.20: Ресунок 20

2.3 Вопросы :

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

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

4. Какие строки листинга 7.4 отвечают за вычисления варианта? О: xor edx,edx mov ebx,20 div ebx inc edx
5. В какой регистр записывается остаток от деления при выполнении инструкции “div ebx”? О:Остаток был записан в регистре edx

6. Для чего используется инструкция “inc edx”? О: Эта инструкция была использована для увеличения значения в регистре edx
7. Какие строки листинга 7.4 отвечают за вывод на экран результата вычислений? О: `mov eax,edx call iprintLF`

2.4 Выводы по результатам выполнения заданий :

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

3 Задание для самостоятельной работы :

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

- Мой вариант : 13
- математическое выражение $(8x + 6) \times 10$
- Итак, мы начали с создания asm-файла, в котором будет находиться наш код.(рис. [-3.1])



```
mmnve@dk8n75:~/work/arch-pc/lab07
mmnve@dk8n75 ~ $ cd ~/work/arch-pc/lab07
mmnve@dk8n75 ~/work/arch-pc/lab07 $ touch test.asm
mmnve@dk8n75 ~/work/arch-pc/lab07 $ ls
in_out.asm  lab7-1.asm.save  lab7-2  lab7-3  test.asm  variant.o
lab7-1      lab7-1.asm.save.1 lab7-2.asm lab7-3.asm variant
lab7-1.asm  lab7-1.o         lab7-2.o  lab7-3.o variant.asm
mmnve@dk8n75 ~/work/arch-pc/lab07 $
```

Рис. 3.1: Ресунок 21

- После этого мы написали код нашей программы. (рис. [-3.2])

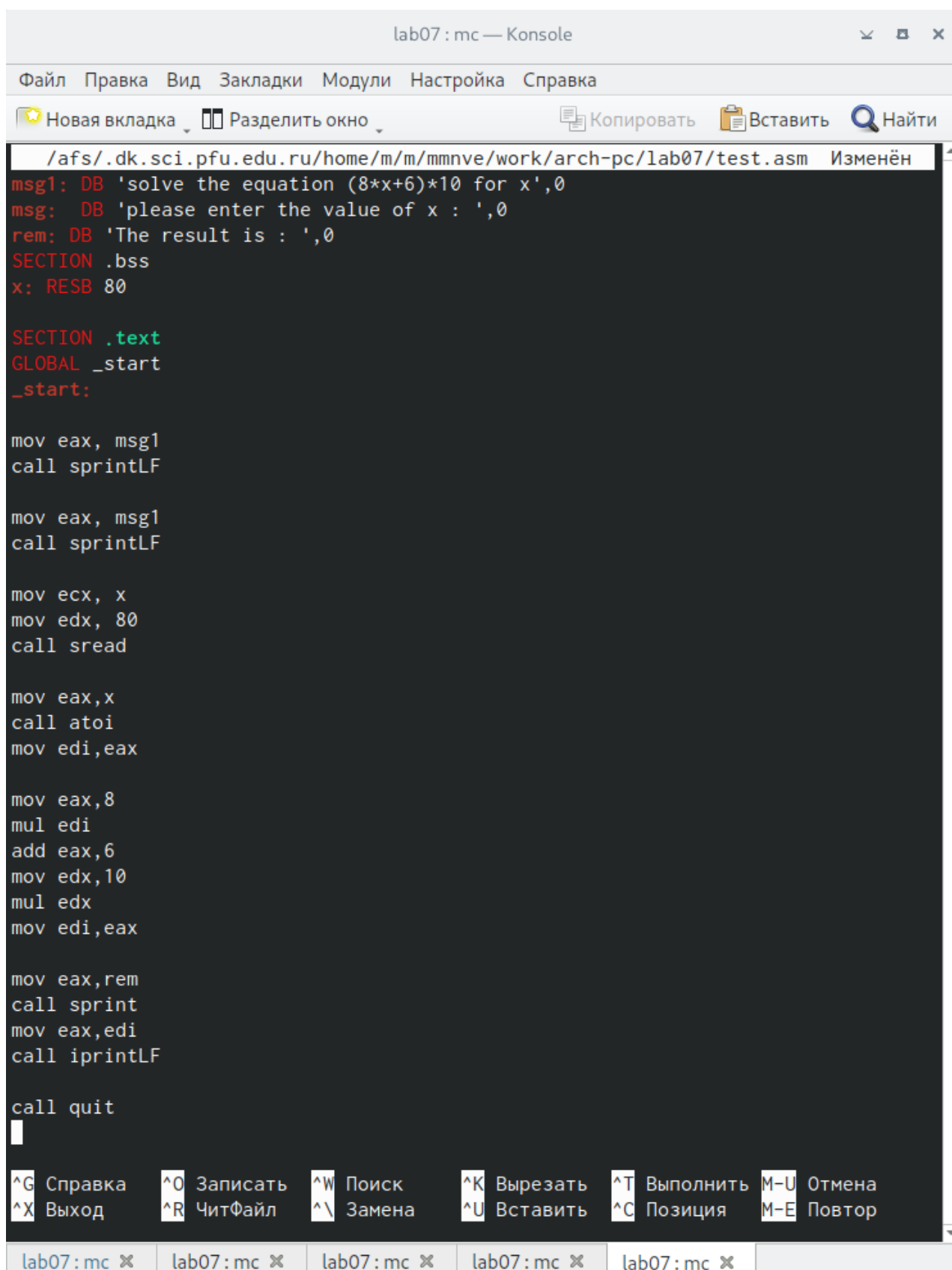
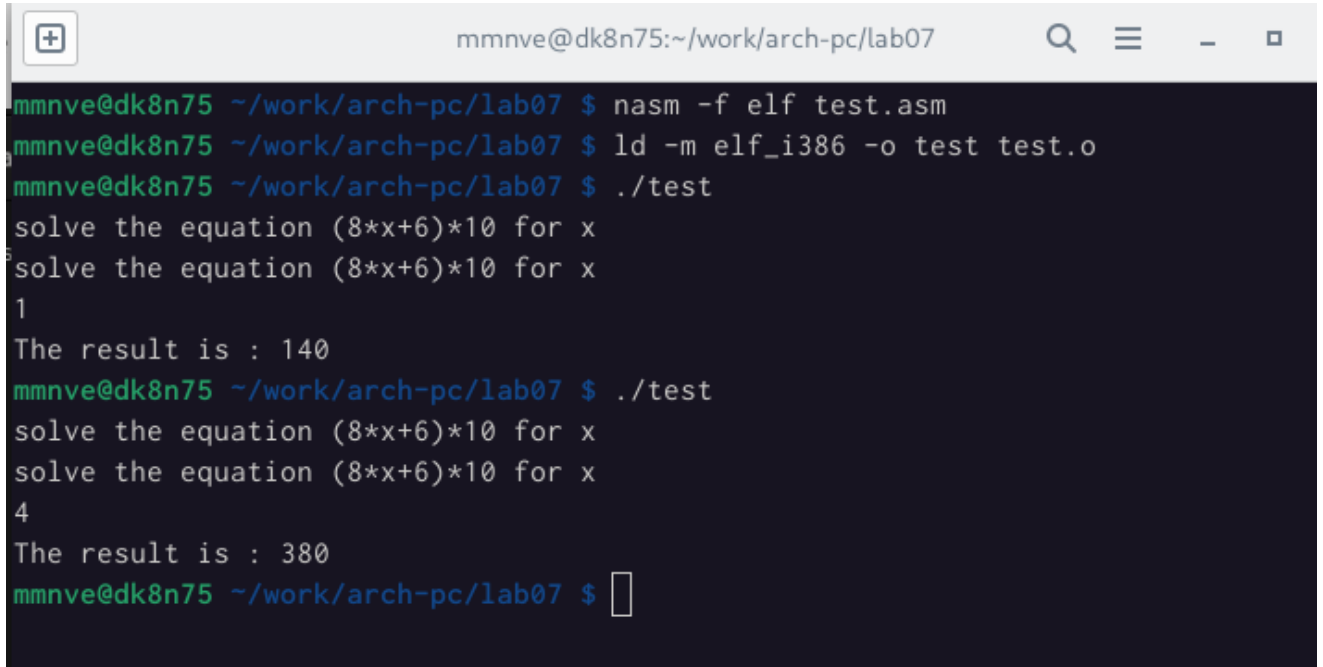


Рис. 3.2: Ресунок 22

- и, наконец, мы проверяем корректность кода, который мы написали, используя два разных значения. $x_1 = 1$ $x_2 = 4$

Как указано на следующем рисунке.(рис. 3.3)



```
mmnve@dk8n75:~/work/arch-pc/lab07
mmnve@dk8n75 ~/work/arch-pc/lab07 $ nasm -f elf test.asm
mmnve@dk8n75 ~/work/arch-pc/lab07 $ ld -m elf_i386 -o test test.o
mmnve@dk8n75 ~/work/arch-pc/lab07 $ ./test
solve the equation (8*x+6)*10 for x
solve the equation (8*x+6)*10 for x
1
The result is : 140
mmnve@dk8n75 ~/work/arch-pc/lab07 $ ./test
solve the equation (8*x+6)*10 for x
solve the equation (8*x+6)*10 for x
4
The result is : 380
mmnve@dk8n75 ~/work/arch-pc/lab07 $
```

Рис. 3.3: Ресунок 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, выполняющие арифметические операции, и научились вычислять математические выражения средней сложности.