

# **Отчёта по лабораторной работе 6**

**Освоение арифметических инструкций языка ассемблера NASM.**

Платонов Максим

# Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Выводы	20

## Список иллюстраций

2.1	Пример программы . . . . .	6
2.2	Работа программы . . . . .	7
2.3	Пример программы . . . . .	7
2.4	Работа программы . . . . .	8
2.5	Пример программы . . . . .	9
2.6	Работа программы . . . . .	9
2.7	Пример программы . . . . .	10
2.8	Работа программы . . . . .	10
2.9	Работа программы . . . . .	11
2.10	Пример программы . . . . .	12
2.11	Работа программы . . . . .	12
2.12	Пример программы . . . . .	13
2.13	Работа программы . . . . .	14
2.14	Пример программы . . . . .	15
2.15	Работа программы . . . . .	16
2.16	Пример программы . . . . .	18
2.17	Работа программы . . . . .	19

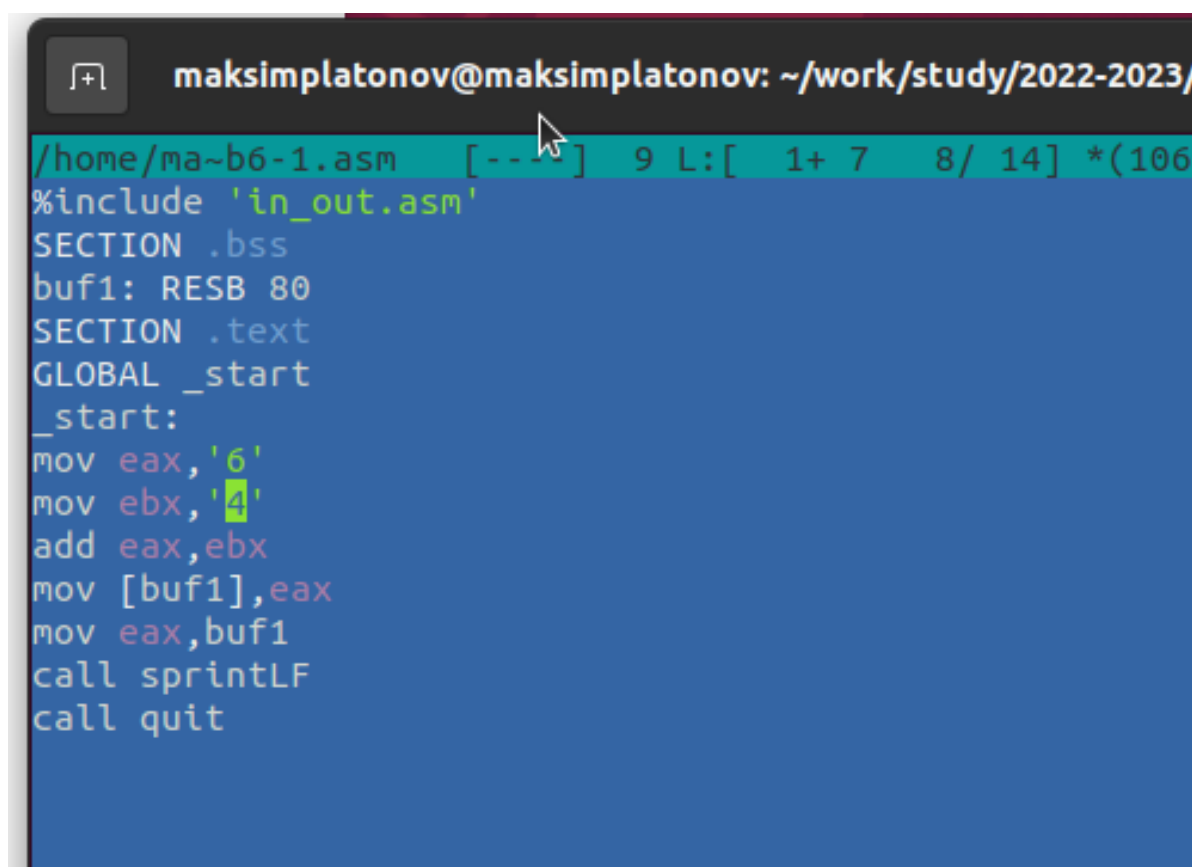
## **Список таблиц**

# 1 Цель работы

Целью работы является освоение арифметических инструкций языка ассемблера NASM.

## 2 Выполнение лабораторной работы

1. Создайте каталог для программ лабораторной работы № 6, перейдите в него и создайте файл lab6-1.asm:
2. Рассмотрим примеры программ вывода символьных и численных значений. Программы будут выводить значения, записанные в регистр eax. (рис. 2.1, 2.2)



```
maksimplatonov@maksimplatonov: ~/work/study/2022-2023/
/home/ma~b6-1.asm [---] 9 L:[ 1+ 7 8/ 14] *(106
#include 'in_out.asm'
SECTION .bss
buf1: RESB 80
SECTION .text
GLOBAL _start
_start:
mov eax, '6'
mov ebx, '4'
add eax, ebx
mov [buf1], eax
mov eax, buf1
call sprintLF
call quit
```

Рис. 2.1: Пример программы

```

maksimplatонов@maksimplatонов:~/work/study/2022-2023/Архитектура компьютера/labs
/lab06$ nasm -f elf lab6-1.asm
maksimplatонов@maksimplatонов:~/work/study/2022-2023/Архитектура компьютера/labs
/lab06$ ld -m elf_i386 -o lab6-1 lab6-1.o
maksimplatонов@maksimplatонов:~/work/study/2022-2023/Архитектура компьютера/labs
/lab06$ ./lab6-1
j
maksimplatонов@maksimplatонов:~/work/study/2022-2023/Архитектура компьютера/labs
/lab06$

```

Рис. 2.2: Работа программы

3. Далее изменим текст программы и вместо символов, запишем в регистры числа. Исправьте текст программы (Листинг 1) следующим образом: (рис. 2.3, 2.4)

```

maksimplatонов@maksimplatонов: ~/work/study/2022-20
/home/ma~b6-1.asm [----] 14 L:[ 1+ 9 10/ 14] *(1
#include 'in_out.asm'
SECTION .bss
buf1: RESB 80
SECTION .text
GLOBAL _start
_start:
mov eax,6
mov ebx,4
add eax,ebx
mov [buf1],eax
mov eax,buf1
call sprintLF
call quit

```

Рис. 2.3: Пример программы

```

/lab06$
maksimlatonov@maksimlatonov:~/work/study/2022-2023/Архитектура компьютера/labs
/lab06$ nasm -f elf lab6-1.asm
maksimlatonov@maksimlatonov:~/work/study/2022-2023/Архитектура компьютера/labs
/lab06$ ld -m elf_i386 -o lab6-1 lab6-1.o
maksimlatonov@maksimlatonov:~/work/study/2022-2023/Архитектура компьютера/labs
/lab06$ ./lab6-1
j
maksimlatonov@maksimlatonov:~/work/study/2022-2023/Архитектура компьютера/labs
/lab06$ nasm -f elf lab6-1.asm
maksimlatonov@maksimlatonov:~/work/study/2022-2023/Архитектура компьютера/labs
/lab06$ ld -m elf_i386 -o lab6-1 lab6-1.o
maksimlatonov@maksimlatonov:~/work/study/2022-2023/Архитектура компьютера/labs
/lab06$ ./lab6-1

maksimlatonov@maksimlatonov:~/work/study/2022-2023/Архитектура компьютера/labs
/lab06$ █

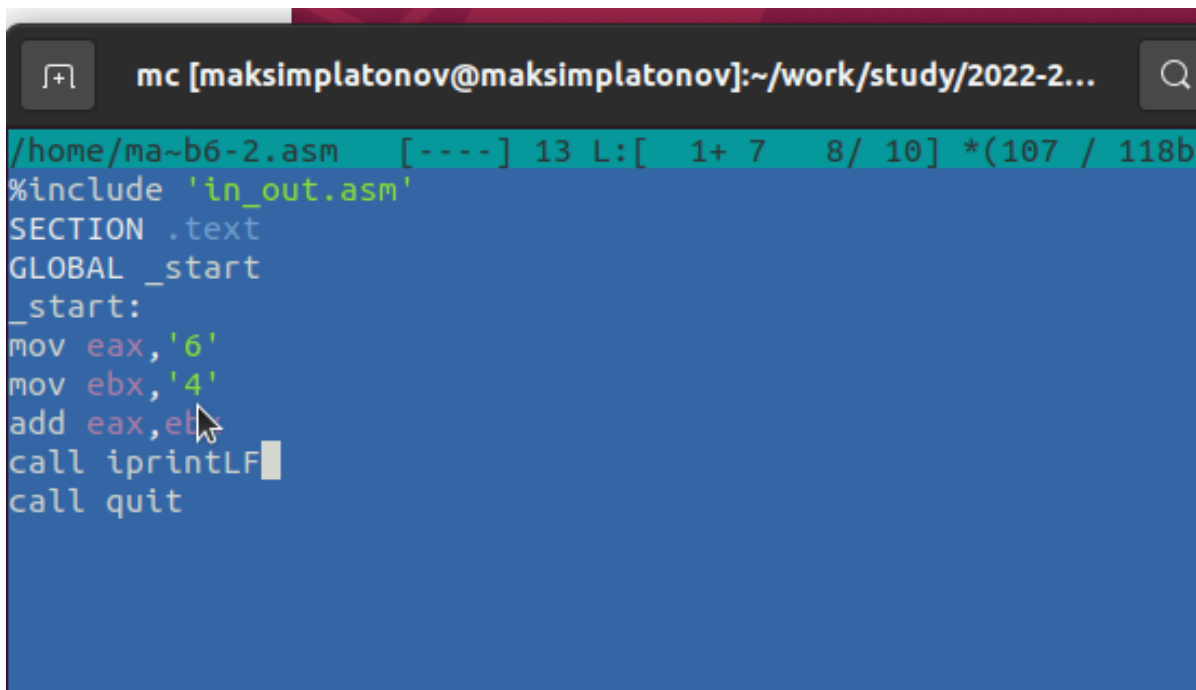
```

Рис. 2.4: Работа программы

Никакой символ не виден, но он есть. Это возврат каретки LF.

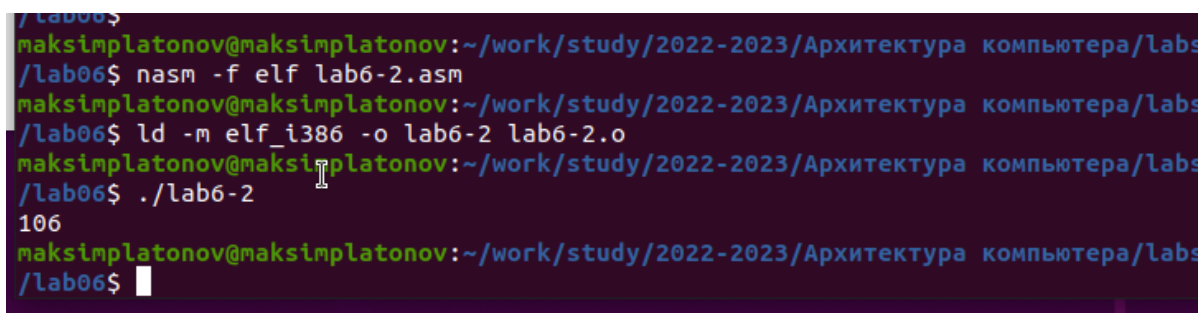
4. Как отмечалось выше, для работы с числами в файле in\_out.asm реализованы подпрограммы для преобразования ASCII символов в числа и обратно. Преобразуем текст программы из Листинга 7.1 с использованием этих функций. (рис. 2.5, 2.6)



A screenshot of a text editor window. The title bar shows the user 'mc [maksimplatonov@maksimplatonov]' and the file path '~/work/study/2022-2...'. The editor displays assembly code for a file named 'lab6-2.asm'. The code includes a header file 'in\_out.asm', sets the text section, and defines a global '\_start' symbol. The '\_start' function moves the ASCII values of '6' and '4' into registers 'eax' and 'ebx' respectively, adds them, and then calls 'iprintLF' to print the result followed by a newline, and 'quit' to exit the program.

```
mc [maksimplatonov@maksimplatonov]:~/work/study/2022-2...
/home/ma~b6-2.asm [----] 13 L:[ 1+ 7 8/ 10] *(107 / 118b
#include 'in_out.asm'
SECTION .text
GLOBAL _start
_start:
mov eax,'6'
mov ebx,'4'
add eax,ebx
call iprintLF
call quit
```

Рис. 2.5: Пример программы

A screenshot of a terminal window showing the compilation and execution of the assembly program. The user is in the directory '/lab06'. They run 'nasm -f elf lab6-2.asm' to assemble the code into an object file 'lab6-2.o'. Then they run 'ld -m elf\_i386 -o lab6-2 lab6-2.o' to link the object file into an executable 'lab6-2'. Finally, they run './lab6-2' to execute the program, which outputs the number '106'.

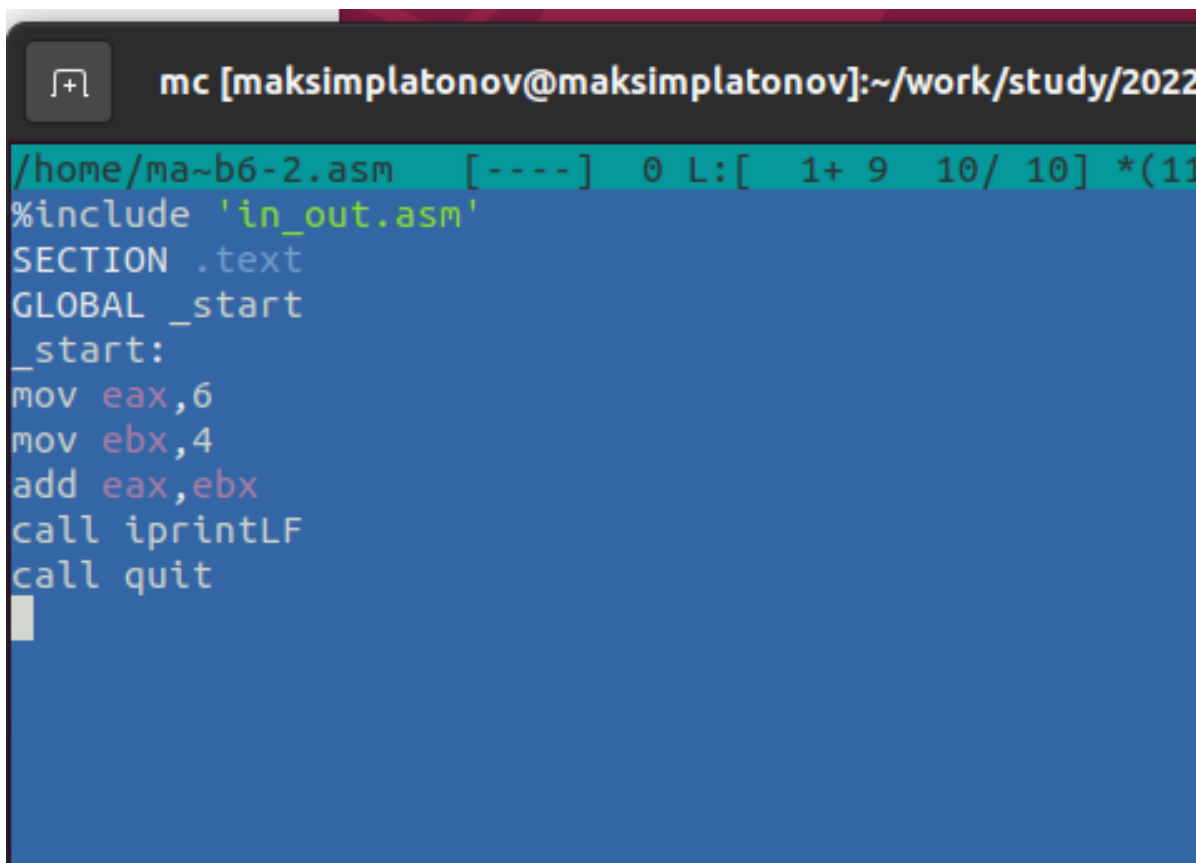
```
/lab06$
maksimplatonov@maksimplatonov:~/work/study/2022-2023/Архитектура компьютера/labs
/lab06$ nasm -f elf lab6-2.asm
maksimplatonov@maksimplatonov:~/work/study/2022-2023/Архитектура компьютера/labs
/lab06$ ld -m elf_i386 -o lab6-2 lab6-2.o
maksimplatonov@maksimplatonov:~/work/study/2022-2023/Архитектура компьютера/labs
/lab06$ ./lab6-2
106
maksimplatonov@maksimplatonov:~/work/study/2022-2023/Архитектура компьютера/labs
/lab06$
```

Рис. 2.6: Работа программы

В результате работы программы мы получим число 106. В данном случае, как и в первом, команда `add` складывает коды символов '6' и '4' ( $54+52=106$ ). Однако, в отличие от программы из листинга 7.1, функция `iprintLF` позволяет вывести число, а не символ, кодом которого является это число.

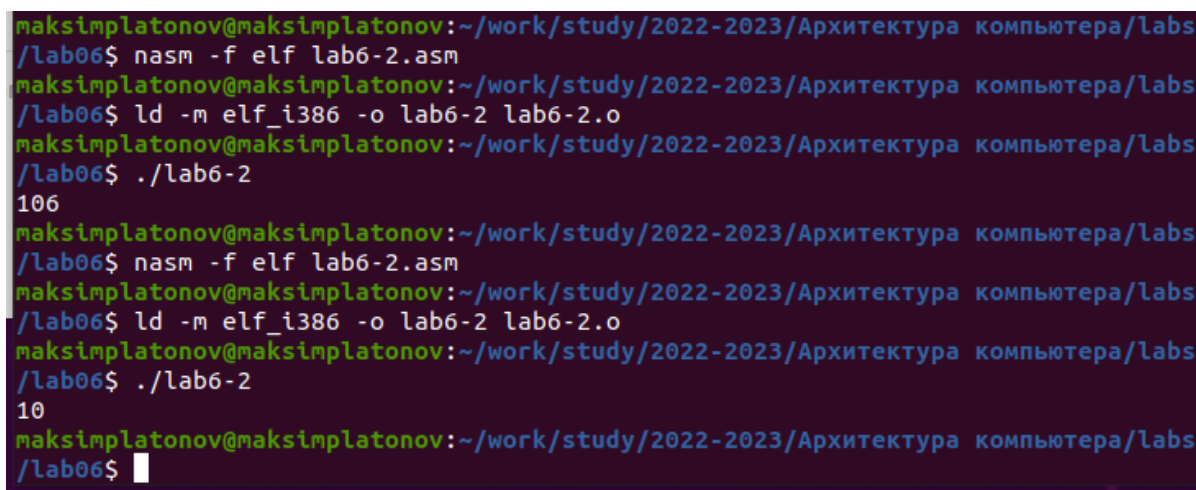
5. Аналогично предыдущему примеру изменим символы на числа. (рис. 2.7, 2.8)

Создайте исполняемый файл и запустите его. Какой результат будет получен при исполнении программы? – получили число 10



```
mc [maksimlatonov@maksimlatonov]:~/work/study/2022-2023/Архитектура компьютера/labs/lab06/
/home/ma~b6-2.asm  [ - - - - ]  0 L: [  1+ 9  10/ 10 ] *(11
%include 'in_out.asm'
SECTION .text
GLOBAL _start
_start:
mov eax,6
mov ebx,4
add eax,ebx
call iprintLF
call quit
```

Рис. 2.7: Пример программы



```
maksimlatonov@maksimlatonov:~/work/study/2022-2023/Архитектура компьютера/labs/lab06$ nasm -f elf lab6-2.asm
maksimlatonov@maksimlatonov:~/work/study/2022-2023/Архитектура компьютера/labs/lab06$ ld -m elf_i386 -o lab6-2 lab6-2.o
maksimlatonov@maksimlatonov:~/work/study/2022-2023/Архитектура компьютера/labs/lab06$ ./lab6-2
106
maksimlatonov@maksimlatonov:~/work/study/2022-2023/Архитектура компьютера/labs/lab06$ nasm -f elf lab6-2.asm
maksimlatonov@maksimlatonov:~/work/study/2022-2023/Архитектура компьютера/labs/lab06$ ld -m elf_i386 -o lab6-2 lab6-2.o
maksimlatonov@maksimlatonov:~/work/study/2022-2023/Архитектура компьютера/labs/lab06$ ./lab6-2
10
maksimlatonov@maksimlatonov:~/work/study/2022-2023/Архитектура компьютера/labs/lab06$
```

Рис. 2.8: Работа программы

Замените функцию `iprintln` на `iprint`. Создайте исполняемый файл и запустите его. Чем отличается вывод функций `iprintln` и `iprint`? - Вывод отличается что нет переноса строки. (рис. 2.9)



```
/lab06$ nasm -f elf lab6-2.asm
maksim@maksimov:~/work/study/2022-2023/Архитектура компьютера/labs
/lab06$ ld -m elf_i386 -o lab6-2 lab6-2.o
maksim@maksimov:~/work/study/2022-2023/Архитектура компьютера/labs
/lab06$ ./lab6-2
106
maksim@maksimov:~/work/study/2022-2023/Архитектура компьютера/labs
/lab06$ nasm -f elf lab6-2.asm
maksim@maksimov:~/work/study/2022-2023/Архитектура компьютера/labs
/lab06$ ld -m elf_i386 -o lab6-2 lab6-2.o
maksim@maksimov:~/work/study/2022-2023/Архитектура компьютера/labs
/lab06$ ./lab6-2
10
maksim@maksimov:~/work/study/2022-2023/Архитектура компьютера/labs
/lab06$ nasm -f elf lab6-2.asm
maksim@maksimov:~/work/study/2022-2023/Архитектура компьютера/labs
/lab06$ ld -m elf_i386 -o lab6-2 lab6-2.o
maksim@maksimov:~/work/study/2022-2023/Архитектура компьютера/labs
/lab06$ ./lab6-2
10
maksim@maksimov:~/work/study/2022-2023/Архитектура компьютера/labs
/lab06$
```

Рис. 2.9: Работа программы

6. В качестве примера выполнения арифметических операций в NASM приведем программу вычисления арифметического выражения

$$f(x) = (5 * 2 + 3) / 3$$

. (рис. 2.10, рис. 2.11)

```
mc [maksimlatonov@maksimlatonov]:~/work/study/2022-2...  
/home/ma~b6-3.asm [----] 9 L:[ 1+13 14/ 27] *(214 /  
%include 'in_out.asm'  
SECTION .data  
div: DB 'Результат: ',0  
rem: DB 'Остаток от деления: ',0  
SECTION .text  
GLOBAL _start  
_start:  
  
mov eax,5  
mov ebx,2  
mul ebx  
add eax,3  
xor edx,edx  
mov ebx,3  
div ebx  
mov edi,eax  
mov eax,div  
call sprint  
mov eax,edi  
call iprintLF  
mov eax,rem  
call sprint  
1 Help 2 Save 3 Mark 4 Replace 5 Copy 6 Move 7 Search
```

Рис. 2.10: Пример программы

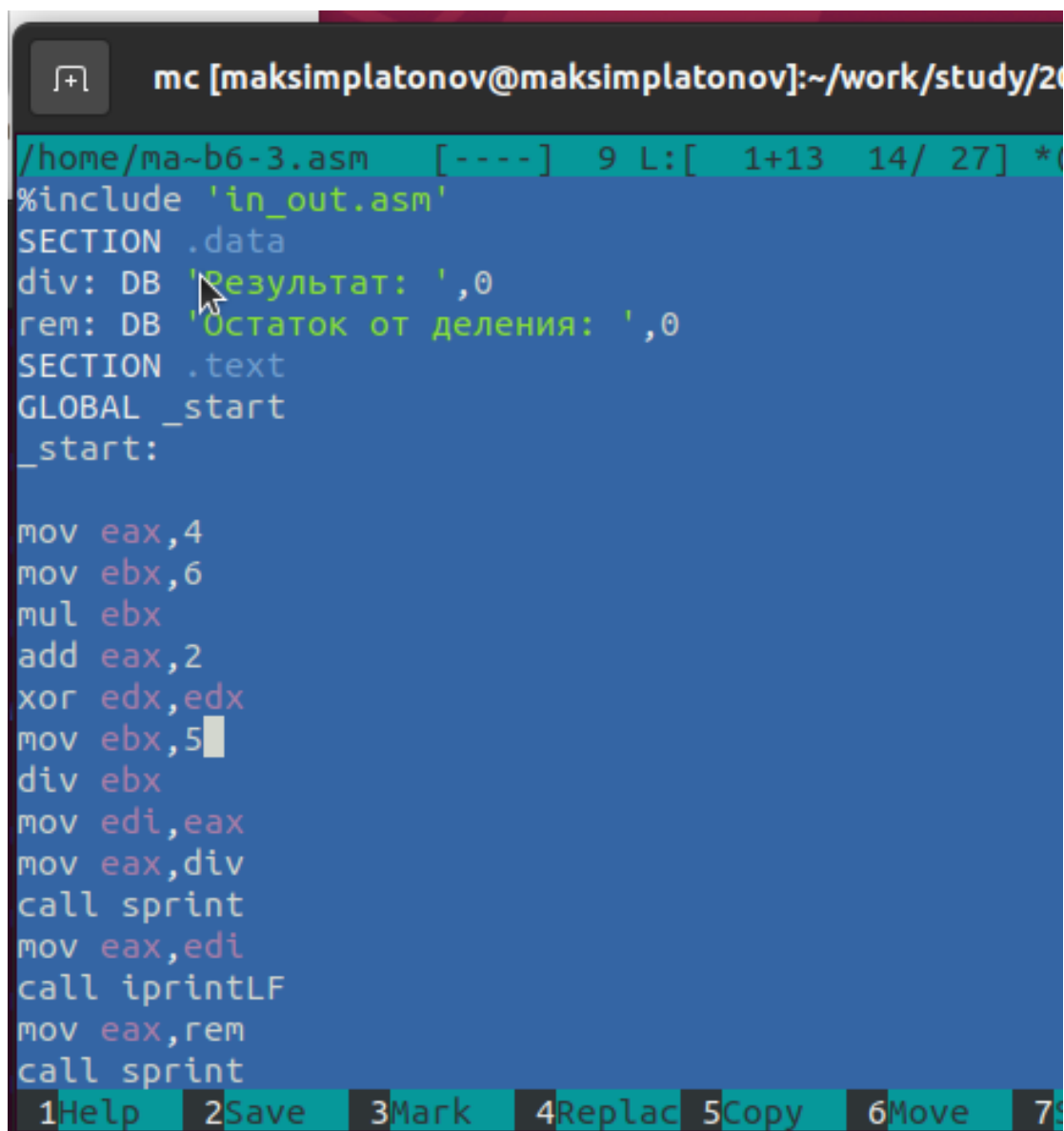
```
maksimlatonov@maksimlatonov:~/work/study/2022-2023/Архитектура компьютера/labs  
/lab06$ nasm -f elf lab6-3.asm  
maksimlatonov@maksimlatonov:~/work/study/2022-2023/Архитектура компьютера/labs  
/lab06$ ld -m elf_i386 -o lab6-3 lab6-3.o  
maksimlatonov@maksimlatonov:~/work/study/2022-2023/Архитектура компьютера/labs  
/lab06$ ./lab6-3  
Результат: 4  
Остаток от деления: 1  
maksimlatonov@maksimlatonov:~/work/study/2022-2023/Архитектура компьютера/labs  
/lab06$
```

Рис. 2.11: Работа программы

Измените текст программы для вычисления выражения

$$f(x) = (4 * 6 + 2) / 5$$

. Создайте исполняемый файл и проверьте его работу. (рис. 2.12, рис. 2.13)



The screenshot shows a text editor window with the title bar "mc [maksimplatonov@maksimplatonov]:~/work/study/20...". The editor contains assembly code for a program that calculates the expression  $f(x) = (4 * 6 + 2) / 5$ . The code is as follows:

```
/home/ma~b6-3.asm [----] 9 L:[ 1+13 14/ 27] *  
%include 'in_out.asm'  
SECTION .data  
div: DB 'Результат: ',0  
rem: DB 'Остаток от деления: ',0  
SECTION .text  
GLOBAL _start  
_start:  
  
mov eax,4  
mov ebx,6  
mul ebx  
add eax,2  
xor edx,edx  
mov ebx,5  
div ebx  
mov edi,eax  
mov eax,div  
call sprint  
mov eax,edi  
call iprintLF  
mov eax,rem  
call sprint
```

At the bottom of the editor, there is a menu bar with the following items: 1Help, 2Save, 3Mark, 4Replac, 5Copy, 6Move, 7S.

Рис. 2.12: Пример программы

```

maksim@maksim:~/work/study/2022-2023/Архитектура компьютера/labs
/lab06$ nasm -f elf lab6-3.asm
maksim@maksim:~/work/study/2022-2023/Архитектура компьютера/labs
/lab06$ ld -m elf_i386 -o lab6-3 lab6-3.o
maksim@maksim:~/work/study/2022-2023/Архитектура компьютера/labs
/lab06$ ./lab6-3
Результат: 4
Остаток от деления: 1
maksim@maksim:~/work/study/2022-2023/Архитектура компьютера/labs
/lab06$ nasm -f elf lab6-3.asm
maksim@maksim:~/work/study/2022-2023/Архитектура компьютера/labs
/lab06$ ld -m elf_i386 -o lab6-3 lab6-3.o
maksim@maksim:~/work/study/2022-2023/Архитектура компьютера/labs
/lab06$ ./lab6-3
Результат: 5
Остаток от деления: 1
maksim@maksim:~/work/study/2022-2023/Архитектура компьютера/labs
/lab06$

```

Рис. 2.13: Работа программы

7. В качестве другого примера рассмотрим программу вычисления варианта задания по номеру студенческого билета, работающую по следующему алгоритму: (рис. 2.14, рис. 2.15)

```
mc [maksimlatonov@maksimlatonov]:~/work/study/2022
/home/ma~iant.asm [----] 0 L:[ 1+26 27/ 27] *(49
%include 'in_out.asm'
SECTION .data
msg: DB 'Введите № 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 ; вызов подпрограммы преобразования
call atoi ; ASCII кода в число, 'eax=x'
xor edx, edx
mov ebx, 20
div ebx
inc edx
mov eax, rem
call sprintf
mov eax, edx
call iprintLF
call quit
```

Рис. 2.14: Пример программы

```

maksim@maksimov:~/work/study/2022-2023/Архитектура компьютера/labs
/lab06$ nasm -f elf variant.asm
maksim@maksimov:~/work/study/2022-2023/Архитектура компьютера/labs
/lab06$ ld -m elf_i386 -o variant variant.o
maksim@maksimov:~/work/study/2022-2023/Архитектура компьютера/labs
/lab06$ ./variant
Введите № студенческого билета:
1132220825
Ваш вариант: 6
maksim@maksimov:~/work/study/2022-2023/Архитектура компьютера/labs
/lab06$

```

Рис. 2.15: Работа программы

- Какие строки листинга 7.4 отвечают за вывод на экран сообщения ‘Ваш вариант:’? – `mov eax,rem` – перекладывает в регистр значение переменной с фразой ‘Ваш вариант:’ `call sprint` – вызов подпрограммы вывода строки
- Для чего используются следующие инструкции? `nasm mov ecx, x mov edx, 80 call sread`

Считывает значение студбилета в переменную X из консоли

- Для чего используется инструкция “`call atoi`”? – эта подпрограмма переводит введенные символы в числовой формат
- Какие строки листинга 7.4 отвечают за вычисления варианта? `xor edx,edx mov ebx,20 div ebx`
- В какой регистр записывается остаток от деления при выполнении инструкции “`div ebx`”? 1 байт AH 2 байта DX 4 байта EDX – наш случай
- Для чего используется инструкция “`inc edx`”? по формуле вычисления варианта нужно прибавить единицу
- Какие строки листинга 7.4 отвечают за вывод на экран результата вычисления `mov eax,edx` – результат перекладывается в регистр `eax call iprintLF` – вызов подпрограммы вывода



8. Написать программу вычисления выражения  $y = f(x)$ . Программа должна выводить выражение для вычисления, выводить запрос на ввод значения  $x$ , вычислять заданное выражение в зависимости от введенного  $x$ , выводить результат вычислений. Вид функции  $f(x)$  выбрать из таблицы 6.3 вариантов заданий в соответствии с номером полученным при выполнении лабораторной работы. Создайте исполняемый файл и проверьте его работу для значений  $x_1$  и  $x_2$  из 6.3. (рис. 2.16, рис. 2.17)

Я получил вариант 6 -

$$x^3/2 + 1$$

для  $x=2$  и  $5$

```
mc [maksimplatnov@maksimplatnov]:~/work/stud
/home/ma~calc.asm [----] 11 L:[ 1+21 22/ 35
%include 'in_out.asm'
SECTION .data
msg: DB 'Введите X ',0
rem: DB 'выражение = : ',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 ; вызов подпрограммы преобразования
call atoi ; ASCII кода в число, `eax=x`

mov ebx,eax
mul ebx
mul ebx

xor edx,edx
mov ebx,2
div ebx
add eax,1

mov ebx,eax
mov eax,rem
call sprintf
```

Рис. 2.16: Пример программы

```
maksimlatonov@maksimlatonov:~/work/study/2022-2023/Архитектура компьютера/labs
/lab06$ nasm -f elf calc.asm
maksimlatonov@maksimlatonov:~/work/study/2022-2023/Архитектура компьютера/labs
/lab06$ ld -m elf_i386 -o calc calc.o
maksimlatonov@maksimlatonov:~/work/study/2022-2023/Архитектура компьютера/labs
/lab06$ ./calc
Введите X
2
выражение = : 5
maksimlatonov@maksimlatonov:~/work/study/2022-2023/Архитектура компьютера/labs
/lab06$ ./calc
Введите X
5
выражение = : 63
maksimlatonov@maksimlatonov:~/work/study/2022-2023/Архитектура компьютера/labs
/lab06$
```

Рис. 2.17: Работа программы

## **3 Выводы**

Изучил работу с арифметическими операциями