

Отчёта по лабораторной работе 6

Освоение арифметических инструкций языка ассемблера NASM.

Галиева Аделина Руслановна

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
4	Выводы	19

Список иллюстраций

Список таблиц

1 Цель работы

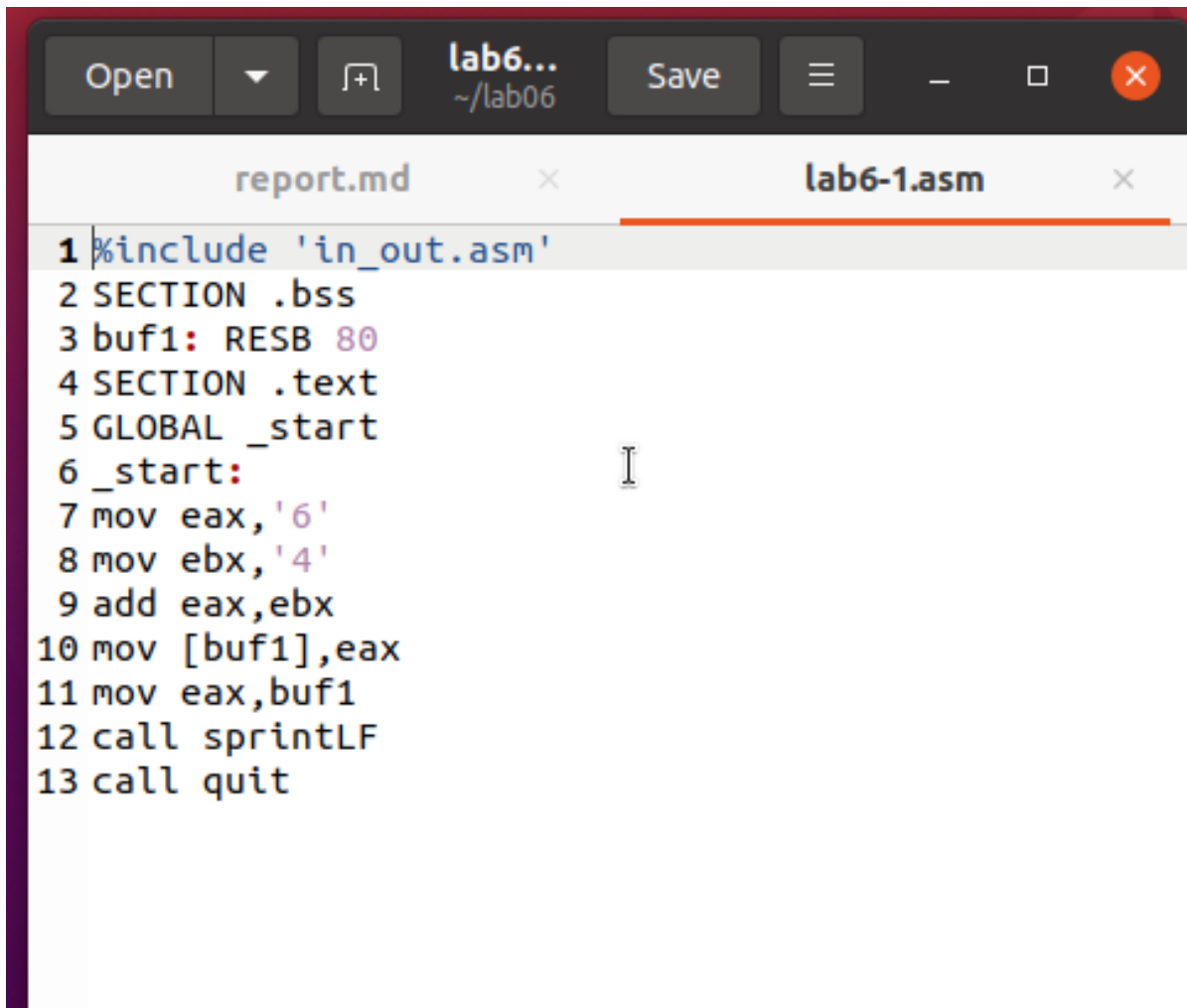
Освоение арифметических инструкций языка ассемблера NASM.

2 Задание

1. Изучить примеры программ.
2. Написать программу вычисления выражения в соответствии с вариантом.
3. Загрузить файлы на GitHub.

3 Выполнение лабораторной работы

1. Создаём каталог для программ лабораторной работы № 6, переходим в него и создаем файл lab7-1.asm:
2. Рассмотрим примеры программ вывода символьных и численных значений. Программы будут выводить значения, записанные в регистр eax.



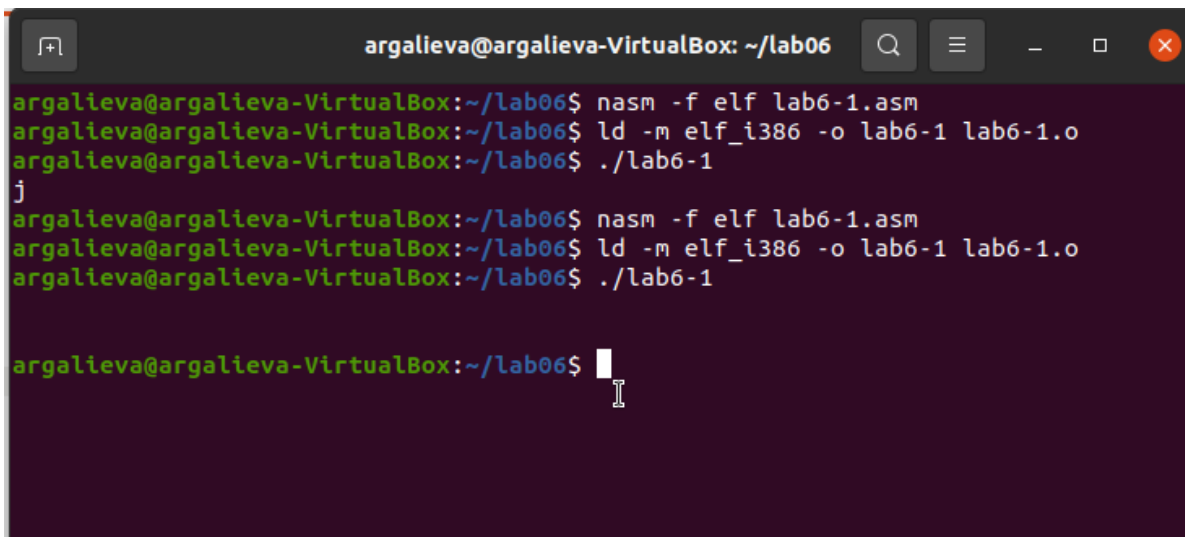
The screenshot shows a code editor window with a dark theme. The title bar includes buttons for 'Open', a dropdown menu, a '+l' icon, the file name 'lab6...' with the path '~/lab06', a 'Save' button, and standard window controls (minimize, maximize, close). The editor has two tabs: 'report.md' and 'lab6-1.asm', with the latter being the active tab. The code in the editor is as follows:

```
1 %include 'in_out.asm'
2 SECTION .bss
3 buf1: RESB 80
4 SECTION .text
5 GLOBAL _start
6 _start:
7 mov eax, '6'
8 mov ebx, '4'
9 add eax, ebx
10 mov [buf1], eax
11 mov eax, buf1
12 call sprintf
13 call quit
```

```
argalleva@argalleva-VirtualBox: ~/lab06
argalleva@argalleva-VirtualBox:~/lab06$ nasm -f elf lab6-1.asm
argalleva@argalleva-VirtualBox:~/lab06$ ld -m elf_i386 -o lab6-1 lab6-1.o
argalleva@argalleva-VirtualBox:~/lab06$ ./lab6-1
j
argalleva@argalleva-VirtualBox:~/lab06$
```

3. Далее меняем текст программы и вместо символов, запишем в регистры числа. Исправляем текст программы (Листинг 1) следующим образом

```
Open lab6... Save
~/lab06
report.md lab6-1.asm
1 %include 'in_out.asm'
2 SECTION .bss
3 buf1: RESB 80
4 SECTION .text
5 GLOBAL _start
6 _start:
7 mov eax,6
8 mov ebx,4
9 add eax,ebx
10 mov [buf1],eax
11 mov eax,buf1
12 call sprintLF
13 call quit
```

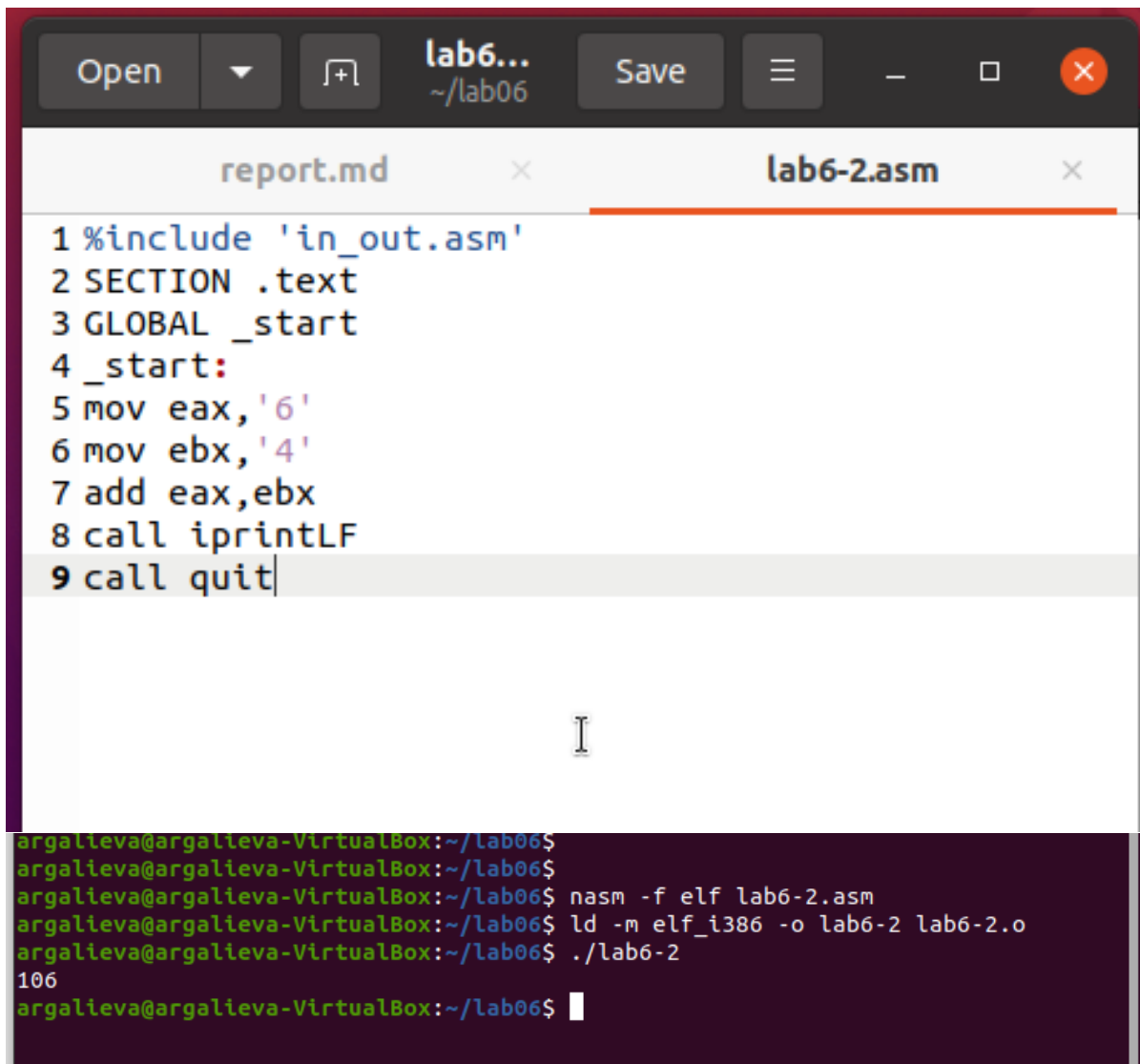

A terminal window titled 'argalieva@argalieva-VirtualBox: ~/lab06'. The window has a dark background with green and blue text. The commands and their outputs are as follows:

```
argalieva@argalieva-VirtualBox:~/lab06$ nasm -f elf lab6-1.asm
argalieva@argalieva-VirtualBox:~/lab06$ ld -m elf_i386 -o lab6-1 lab6-1.o
argalieva@argalieva-VirtualBox:~/lab06$ ./lab6-1
j
argalieva@argalieva-VirtualBox:~/lab06$ nasm -f elf lab6-1.asm
argalieva@argalieva-VirtualBox:~/lab06$ ld -m elf_i386 -o lab6-1 lab6-1.o
argalieva@argalieva-VirtualBox:~/lab06$ ./lab6-1

argalieva@argalieva-VirtualBox:~/lab06$ █
```

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.



The screenshot shows a code editor window with two tabs: 'report.md' and 'lab6-2.asm'. The 'lab6-2.asm' tab is active, displaying the following assembly code:

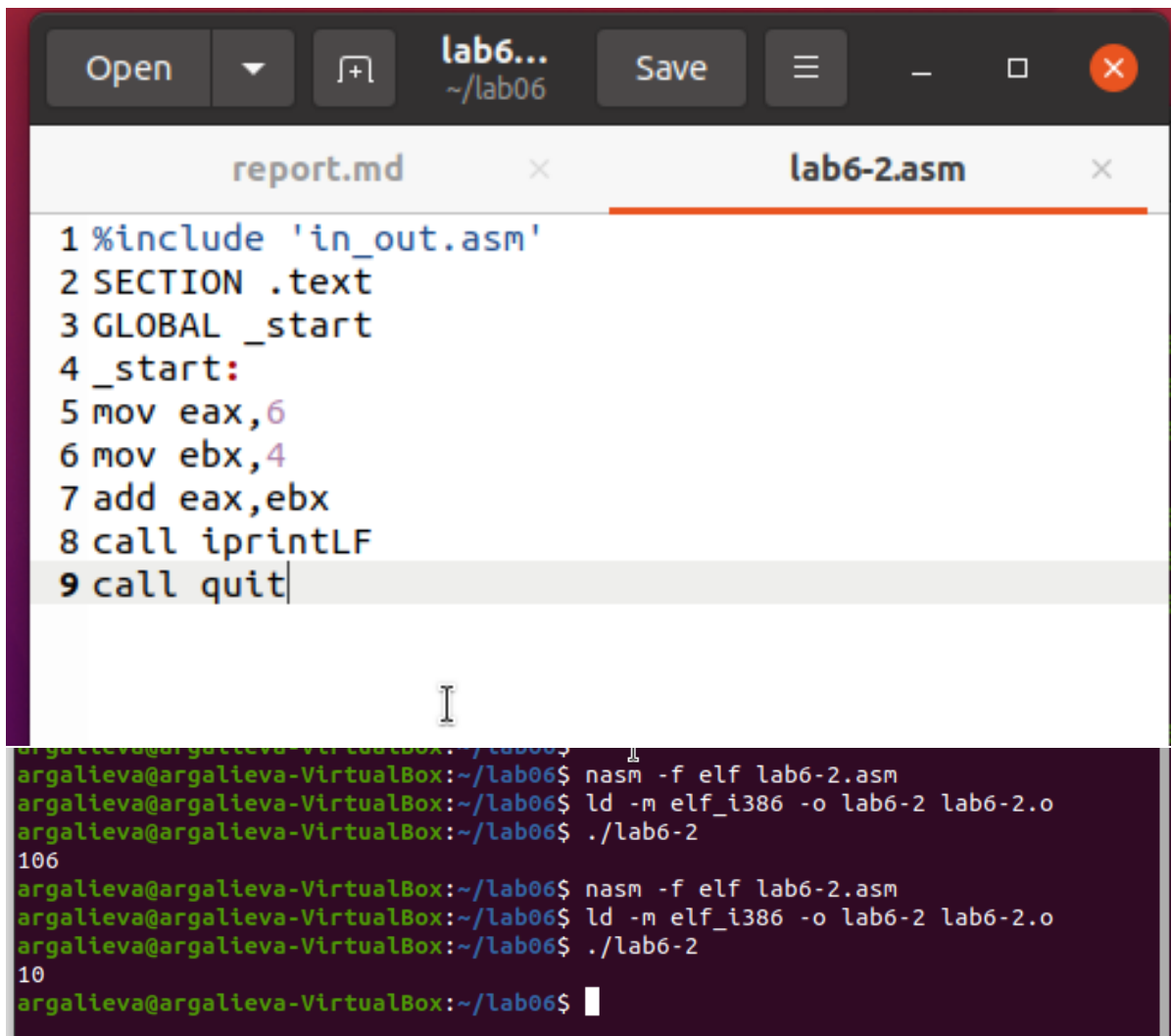
```
1 %include 'in_out.asm'
2 SECTION .text
3 GLOBAL _start
4 _start:
5 mov eax, '6'
6 mov ebx, '4'
7 add eax, ebx
8 call iprintLF
9 call quit
```

Below the code editor is a terminal window with the following commands and output:

```
argalleva@argalleva-VirtualBox:~/lab06$
argalleva@argalleva-VirtualBox:~/lab06$
argalleva@argalleva-VirtualBox:~/lab06$ nasm -f elf lab6-2.asm
argalleva@argalleva-VirtualBox:~/lab06$ ld -m elf_i386 -o lab6-2 lab6-2.o
argalleva@argalleva-VirtualBox:~/lab06$ ./lab6-2
106
argalleva@argalleva-VirtualBox:~/lab06$
```

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. Создаём исполняемый файл и запускаем его. Какой результат будет получен при исполнении программы? – получим число 10



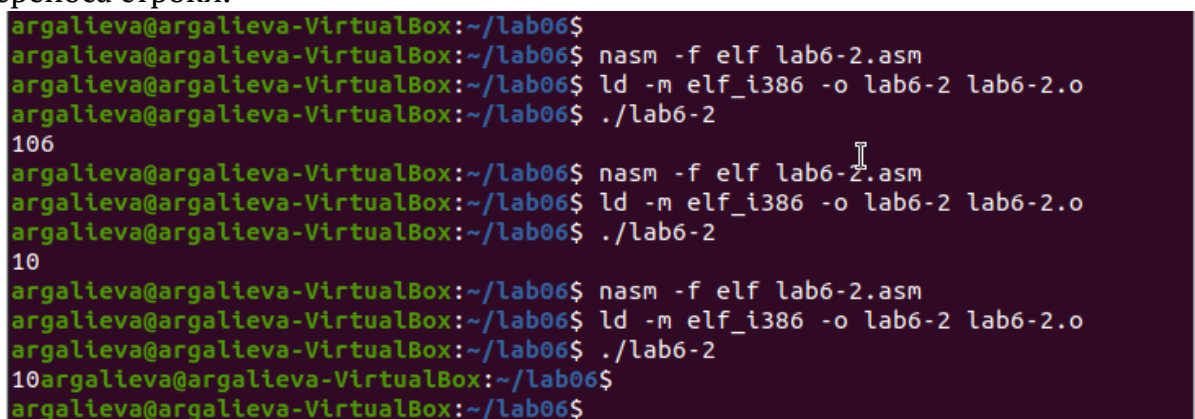
The screenshot shows a code editor window with two tabs: 'report.md' and 'lab6-2.asm'. The 'lab6-2.asm' tab is active and contains the following assembly code:

```
1 %include 'in_out.asm'
2 SECTION .text
3 GLOBAL _start
4 _start:
5 mov eax,6
6 mov ebx,4
7 add eax,ebx
8 call iprintLF
9 call quit
```

Below the code editor is a terminal window showing the compilation process:

```
argaliev@argaliev-VirtualBox:~/lab06$ nasm -f elf lab6-2.asm
argaliev@argaliev-VirtualBox:~/lab06$ ld -m elf_i386 -o lab6-2 lab6-2.o
argaliev@argaliev-VirtualBox:~/lab06$ ./lab6-2
106
argaliev@argaliev-VirtualBox:~/lab06$ nasm -f elf lab6-2.asm
argaliev@argaliev-VirtualBox:~/lab06$ ld -m elf_i386 -o lab6-2 lab6-2.o
argaliev@argaliev-VirtualBox:~/lab06$ ./lab6-2
10
argaliev@argaliev-VirtualBox:~/lab06$
```

Заменяем функцию `iprintLF` на `iprint`. Создаем исполняемый файл и запускаем его. Чем отличается вывод функций `iprintLF` и `iprint`? - Вывод отличается что нет переноса строки.

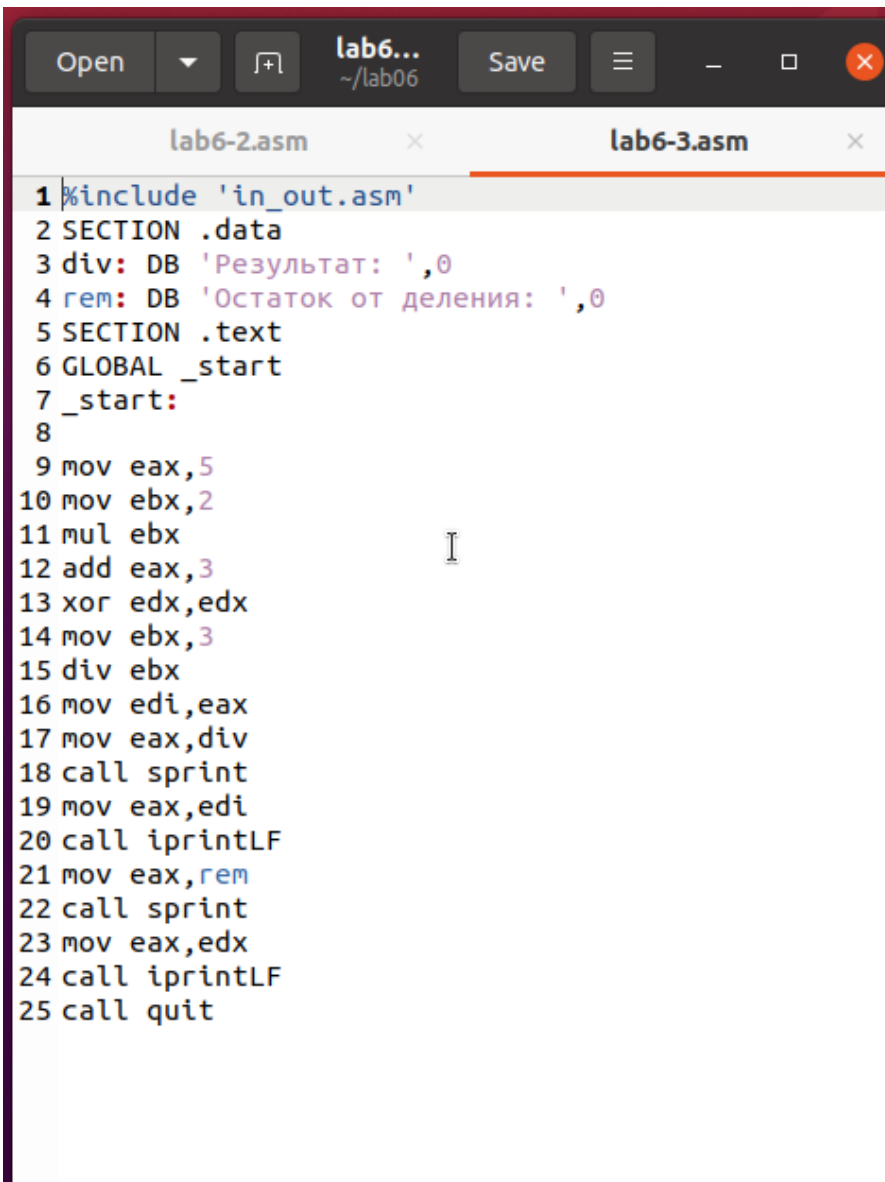


The screenshot shows a terminal window with the following commands and output:

```
argaliev@argaliev-VirtualBox:~/lab06$
argaliev@argaliev-VirtualBox:~/lab06$ nasm -f elf lab6-2.asm
argaliev@argaliev-VirtualBox:~/lab06$ ld -m elf_i386 -o lab6-2 lab6-2.o
argaliev@argaliev-VirtualBox:~/lab06$ ./lab6-2
106
argaliev@argaliev-VirtualBox:~/lab06$ nasm -f elf lab6-2.asm
argaliev@argaliev-VirtualBox:~/lab06$ ld -m elf_i386 -o lab6-2 lab6-2.o
argaliev@argaliev-VirtualBox:~/lab06$ ./lab6-2
10
argaliev@argaliev-VirtualBox:~/lab06$ nasm -f elf lab6-2.asm
argaliev@argaliev-VirtualBox:~/lab06$ ld -m elf_i386 -o lab6-2 lab6-2.o
argaliev@argaliev-VirtualBox:~/lab06$ ./lab6-2
10
argaliev@argaliev-VirtualBox:~/lab06$
```

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

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



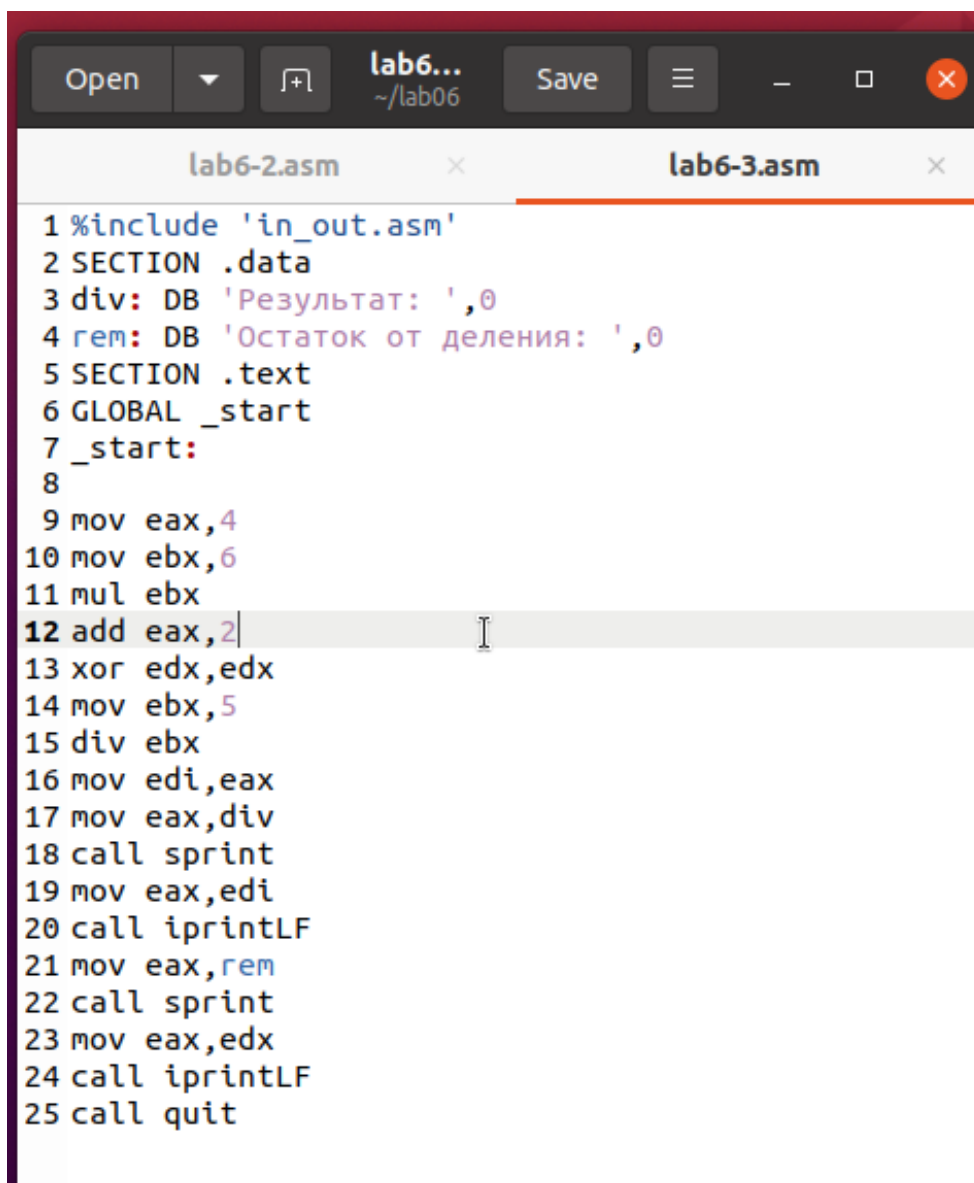
```
1 %include 'in_out.asm'
2 SECTION .data
3 div: DB 'Результат: ',0
4 rem: DB 'Остаток от деления: ',0
5 SECTION .text
6 GLOBAL _start
7 _start:
8
9 mov eax,5
10 mov ebx,2
11 mul ebx
12 add eax,3
13 xor edx,edx
14 mov ebx,3
15 div ebx
16 mov edi,eax
17 mov eax,div
18 call sprint
19 mov eax,edi
20 call iprintLF
21 mov eax,rem
22 call sprint
23 mov eax,edx
24 call iprintLF
25 call quit
```

```
argaliev@argaliev-VirtualBox:~/lab06$  
argaliev@argaliev-VirtualBox:~/lab06$  
argaliev@argaliev-VirtualBox:~/lab06$ nasm -f elf lab6-3.asm  
argaliev@argaliev-VirtualBox:~/lab06$ ld -m elf_i386 -o lab6-3 lab6-3.o  
argaliev@argaliev-VirtualBox:~/lab06$ ./lab6-3  
Результат: 4  
Остаток от деления: 1  
argaliev@argaliev-VirtualBox:~/lab06$  
argaliev@argaliev-VirtualBox:~/lab06$
```

Изменяем текст программы для вычисления выражения

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

. Создаем исполняемый файл и проверяем его работу.



The image shows a code editor window with a dark theme. The title bar at the top contains the text "lab6..." followed by a tilde and "lab06". To the left of the title bar are buttons for "Open", a dropdown arrow, and a "Save" button. To the right are window control buttons (minimize, maximize, close). Below the title bar, there are two tabs: "lab6-2.asm" and "lab6-3.asm", with the latter being the active tab. The main area of the editor displays assembly code. The code starts with a line number 1 and includes a directive to include "in_out.asm". It then defines two data sections: ".data" and ".text". In the ".data" section, it defines two variables: "div" and "rem", both of type "DB" (define byte). The ".text" section is marked as "GLOBAL _start". The code then defines the "_start" label and begins the main logic. It moves the value 4 into "eax", then 6 into "ebx", and multiplies them. Line 12, which is highlighted, adds 2 to "eax". Subsequent lines perform a division of "eax" by "ebx", store the quotient in "edi", and call "sprintf" to format the result. It then prints the result using "iprintLF". The remainder is stored in "eax", and the code calls "sprintf" and "iprintLF" again to display it. Finally, it calls "quit" to end the program.

```
1 %include 'in_out.asm'
2 SECTION .data
3 div: DB 'Результат: ',0
4 rem: DB 'Остаток от деления: ',0
5 SECTION .text
6 GLOBAL _start
7 _start:
8
9 mov eax,4
10 mov ebx,6
11 mul ebx
12 add eax,2
13 xor edx,edx
14 mov ebx,5
15 div ebx
16 mov edi,eax
17 mov eax,div
18 call sprintf
19 mov eax,edi
20 call iprintLF
21 mov eax,rem
22 call sprintf
23 mov eax,edx
24 call iprintLF
25 call quit
```

```

argaliev@argaliev-VirtualBox:~/lab06$
argaliev@argaliev-VirtualBox:~/lab06$
argaliev@argaliev-VirtualBox:~/lab06$ nasm -f elf lab6-3.asm
argaliev@argaliev-VirtualBox:~/lab06$ ld -m elf_i386 -o lab6-3 lab6-3.o
argaliev@argaliev-VirtualBox:~/lab06$ ./lab6-3
Результат: 4
Остаток от деления: 1
argaliev@argaliev-VirtualBox:~/lab06$
argaliev@argaliev-VirtualBox:~/lab06$
argaliev@argaliev-VirtualBox:~/lab06$
argaliev@argaliev-VirtualBox:~/lab06$
argaliev@argaliev-VirtualBox:~/lab06$ nasm -f elf lab6-3.asm
argaliev@argaliev-VirtualBox:~/lab06$ ld -m elf_i386 -o lab6-3 lab6-3.o
argaliev@argaliev-VirtualBox:~/lab06$ ./lab6-3
Результат: 5
Остаток от деления: 1
argaliev@argaliev-VirtualBox:~/lab06$
argaliev@argaliev-VirtualBox:~/lab06$
argaliev@argaliev-VirtualBox:~/lab06$

```

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

The image shows a code editor window with a dark theme. The title bar indicates the file is named 'vari...' and is located in the directory '~/lab06'. The editor contains assembly code for a program that prompts the user for a student ticket number and a variant number. The code includes sections for data, bss, and text, and uses various assembly instructions like `mov`, `call`, `sprint`, and `iprintLF`. A cursor is visible on line 22.

```
1 %include 'in_out.asm'
2 SECTION .data
3 msg: DB 'Введите № студенческого билета: ',0
4 rem: DB 'Ваш вариант: ',0
5 SECTION .bss
6 x: RESB 80
7 SECTION .text
8 GLOBAL _start
9 _start:
10 mov eax, msg
11 call sprintLF
12 mov ecx, x
13 mov edx, 80
14 call sread
15 mov eax,x ; вызов подпрограммы преобразования
16 call atoi ; ASCII кода в число, 'eax=x'
17 xor edx,edx
18 mov ebx,20
19 div ebx
20 inc edx
21 mov eax,rem
22 call sprint
23 mov eax,edx
24 call iprintLF
25 call quit
26 |
```

Below the editor is a terminal window with a dark background. It shows the execution of the assembly code using `nasm` and `ld`, followed by the program's output prompts and user input.

```
argalieva@argalieva-VirtualBox:~/lab06$
argalieva@argalieva-VirtualBox:~/lab06$ nasm -f elf variant.asm
argalieva@argalieva-VirtualBox:~/lab06$ ld -m elf_i386 -o variant variant.o
argalieva@argalieva-VirtualBox:~/lab06$ ./variant
Введите № студенческого билета:
1132229058
Ваш вариант: 19
argalieva@argalieva-VirtualBox:~/lab06$
argalieva@argalieva-VirtualBox:~/lab06$
```

- 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? `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 вычислений `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. Создаем исполняемый файл и проверяем его работу для значений x_1 и x_2 из 6.3.

Вариант 19 -

$$(x/3 + 5) * 7$$

для $x=3$ и 9

{ #fig:016 width=70%, height=70% }

```
argaliev@argaliev-VirtualBox: ~/lab06$  
argaliev@argaliev-VirtualBox:~/lab06$  
argaliev@argaliev-VirtualBox:~/lab06$ nasm -f elf calc.asm  
argaliev@argaliev-VirtualBox:~/lab06$ ld -m elf_i386 -o calc calc.o  
argaliev@argaliev-VirtualBox:~/lab06$ ./calc  
Введите X  
3  
выражение = : 42  
argaliev@argaliev-VirtualBox:~/lab06$ ./calc  
Введите X  
9  
выражение = : 56  
argaliev@argaliev-VirtualBox:~/lab06$
```

4 Выводы

Я изучила работу с арифметическими операциями.