

Отчёта по лабораторной работе №6

Освоение арифметических инструкций языка ассемблера NASM.

Гусейнов Тагир Гамзатович

Содержание

1 Цель работы

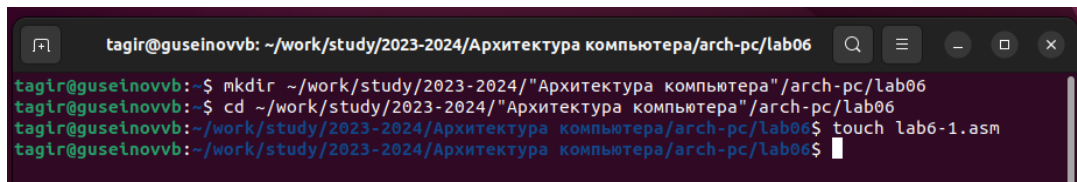
Целью работы является освоение арифметических инструкций языка ассемблера NASM.

2 Задание

1. Написать программу вычисления выражения $y = f(x)$. Вид функции $f(x)$ выбрать из таблицы 6.3 вариантов заданий в соответствии с номером полученным при выполнении лабораторной работы.
2. Загрузите файлы на GitHub.

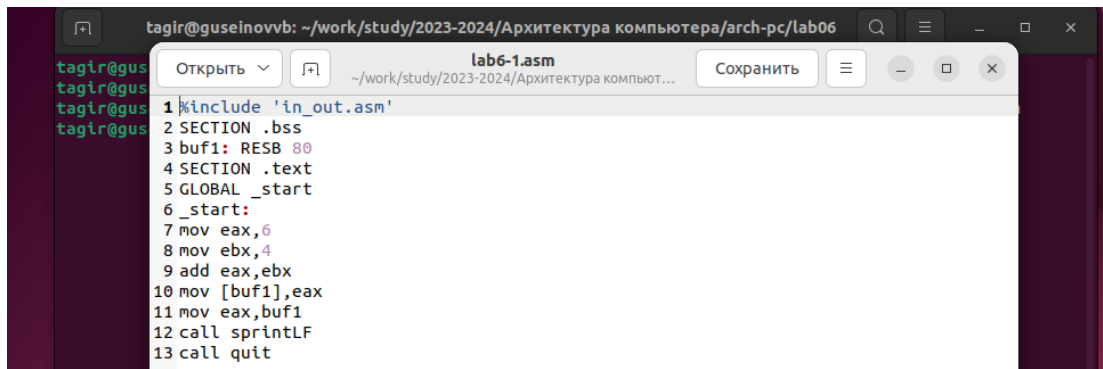
3 Выполнение лабораторной работы

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. 1, 2)



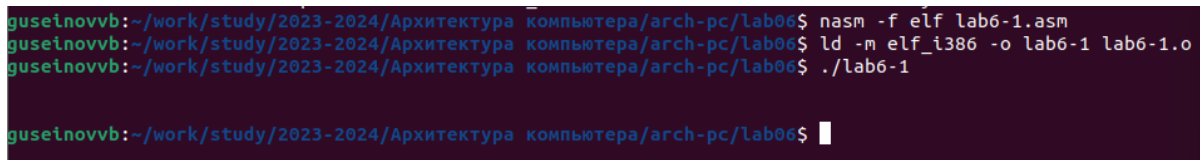
```
tagir@guseinovvb: ~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab06
tagir@guseinovvb:~$ mkdir ~/work/study/2023-2024/"Архитектура компьютера"/arch-pc/lab06
tagir@guseinovvb:~$ cd ~/work/study/2023-2024/"Архитектура компьютера"/arch-pc/lab06
tagir@guseinovvb:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab06$ touch lab6-1.asm
tagir@guseinovvb:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab06$
```

Рис 1. создание лаб 6-1



```
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
```

Рис. 2 программа 1

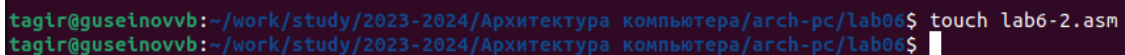


```
guseinovvb:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab06$ nasm -f elf lab6-1.asm
guseinovvb:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab06$ ld -m elf_i386 -o lab6-1 lab6-1.o
guseinovvb:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab06$ ./lab6-1
```

Рис. 3: Запуск лаб 6-1

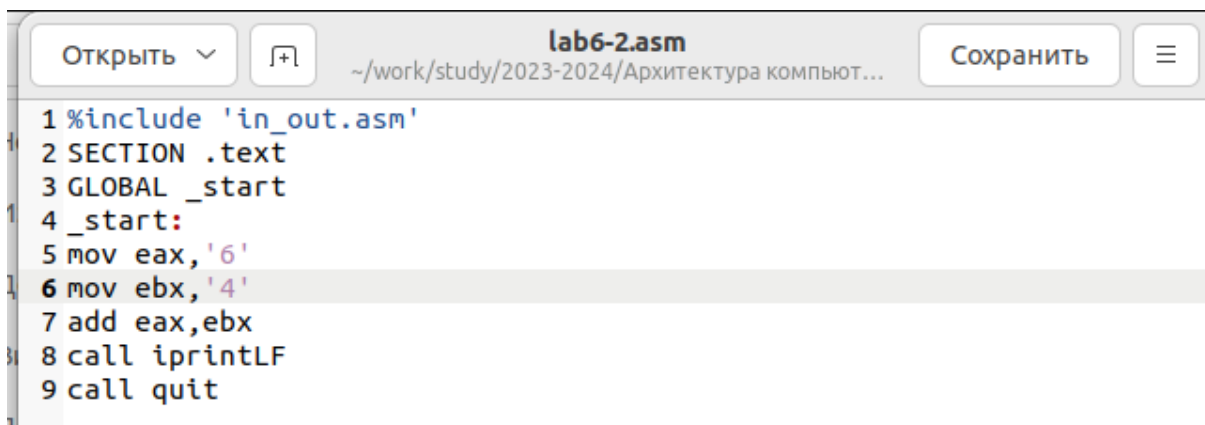
Никакой символ не виден, но он есть. Это возврат каретки LF.

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. 5, 6)



```
tagir@guseinovvb:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab06$ touch lab6-2.asm
tagir@guseinovvb:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab06$
```

Рис 4 создание лаб 6 –2



```
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
```

Рис. 5: программа лаб 6 –2

```

tagir@guseinovvb:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab06$ nasm -f elf lab6-2.asm
tagir@guseinovvb:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab06$ ld -m elf_i386 -o lab6-2 lab6-2.o
tagir@guseinovvb:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab06$ ./lab6-2
106
tagir@guseinovvb:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab06$

```

Рис. 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. 7, 8)

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



```

lab6-2.asm
~/work/study/2023-2024/Архитектура компьют...
Открыть  [+/-] Сохранить

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

```

Рис. 7: Пример программы

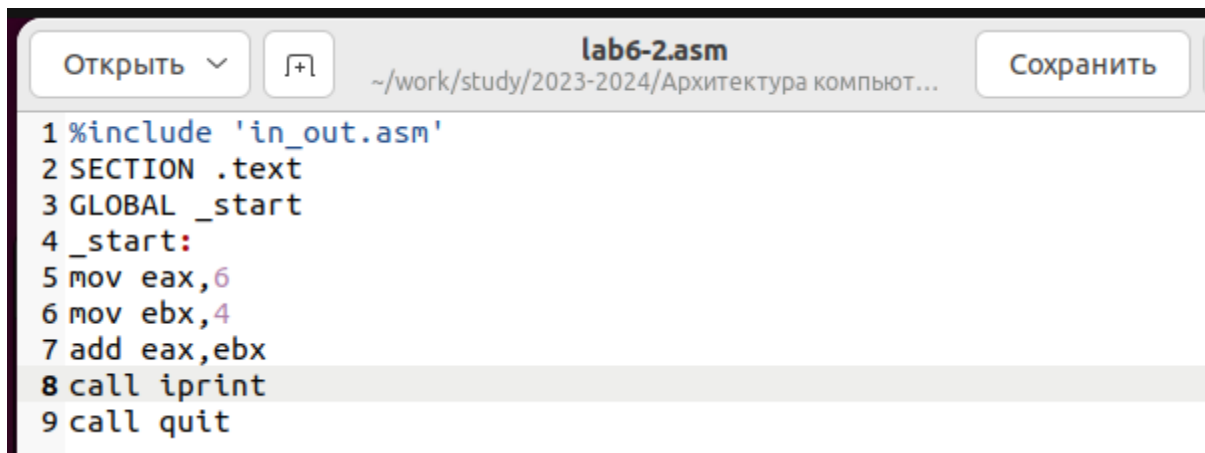
```

tagir@guseinovvb:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab06$ nasm -f elf lab6-2.asm
tagir@guseinovvb:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab06$ ld -m elf_i386 -o lab6-2 lab6-2.o
tagir@guseinovvb:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab06$ ./lab6-2
10
tagir@guseinovvb:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab06$

```

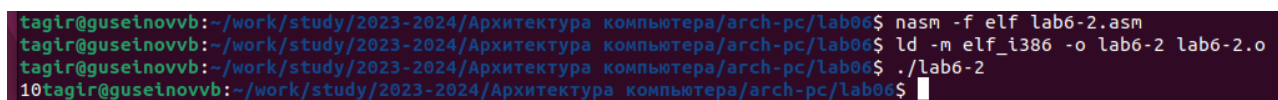
Рис. 8: Работа программы

Замените функцию `iprintLF` на `iprint`. Создайте исполняемый файл и запустите его. Чем отличается вывод функций `iprintLF` и `iprint`? - Вывод отличается что нет переноса строки. (рис. 9)



```
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 iprint
9 call quit
```

Рис 9 Измененная лаб 6 –2 (2)

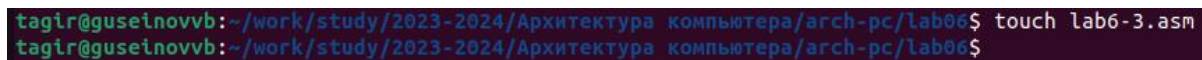


```
tagir@guseinovvb:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab06$ nasm -f elf lab6-2.asm
tagir@guseinovvb:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab06$ ld -m elf_i386 -o lab6-2 lab6-2.o
tagir@guseinovvb:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab06$ ./lab6-2
10tagir@guseinovvb:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab06$
```

Рис. 10: Работа программы

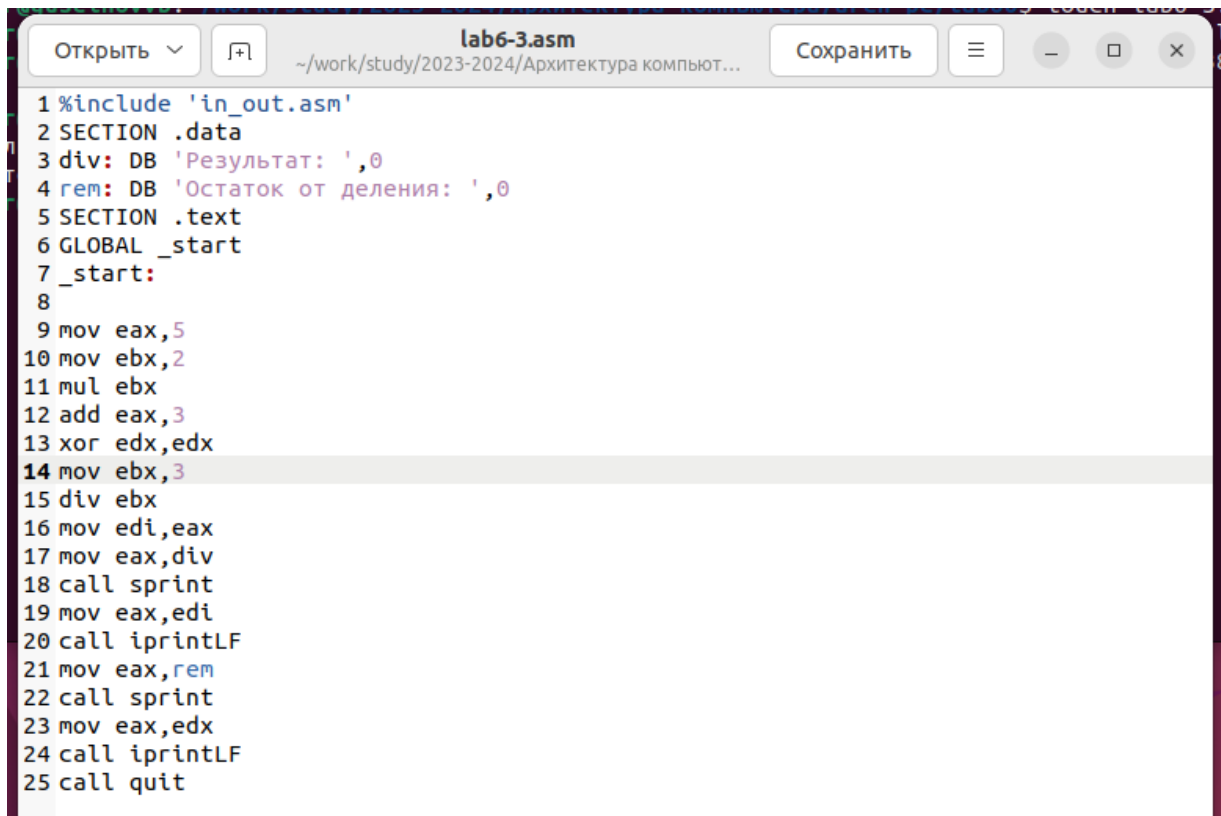
6. В качестве примера выполнения арифметических операций в NASM приведем программу вычисления арифметического выражения

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



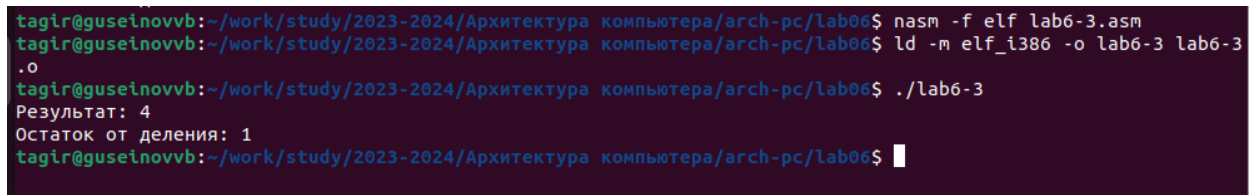
```
tagir@guseinovvb:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab06$ touch lab6-3.asm
tagir@guseinovvb:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab06$
```

Рис 11, создание лаб 6-3



```
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
```

Рис. 12: программа лаб 6-3



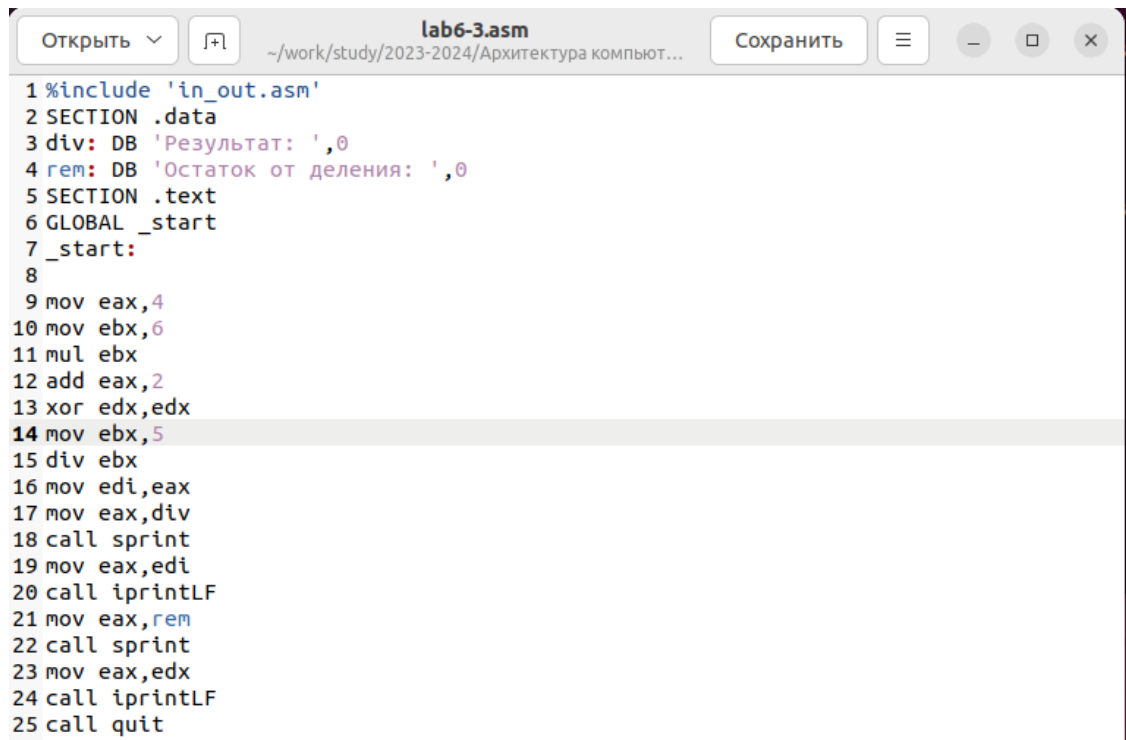
```
tagir@guseinovvb:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab06$ nasm -f elf lab6-3.asm
tagir@guseinovvb:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab06$ ld -m elf_i386 -o lab6-3 lab6-3.o
tagir@guseinovvb:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab06$ ./lab6-3
Результат: 4
Остаток от деления: 1
tagir@guseinovvb:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab06$
```

Рис. 13, работа лаб 6-3

Измените текст программы для вычисления выражения

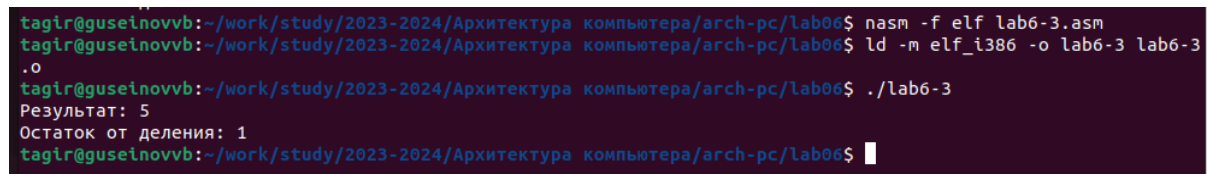
$$f(x) = (4 * 6 + 2)/5$$

. Создайте исполняемый файл и проверьте его работу.



```
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 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
```

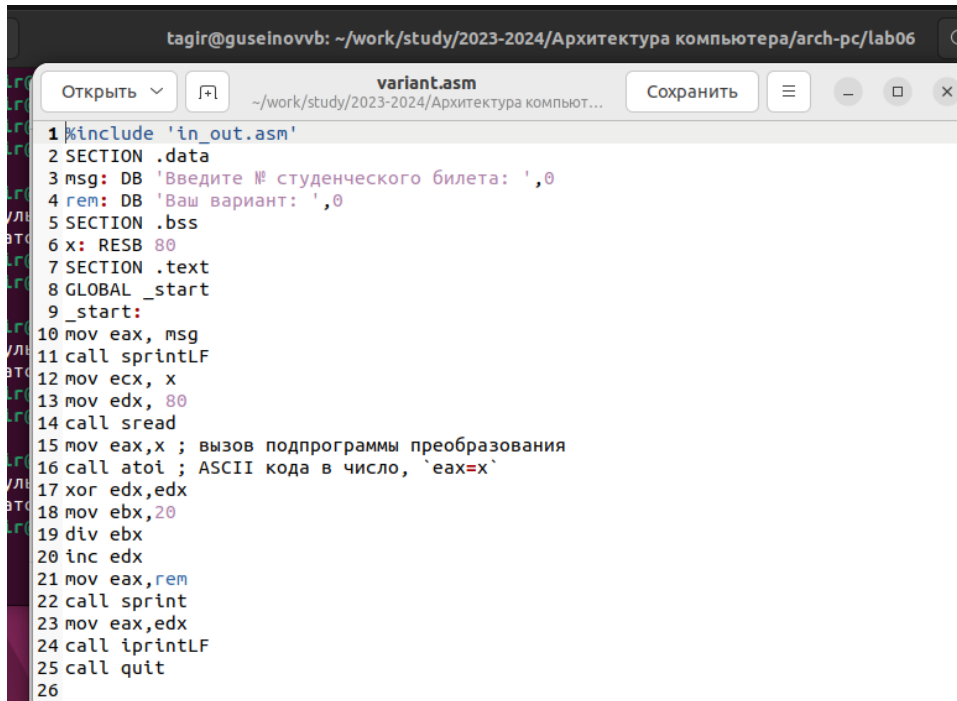
Рис. 14. программа лаб 6-3 в2



```
tagir@guseinovvb:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab06$ nasm -f elf lab6-3.asm
tagir@guseinovvb:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab06$ ld -m elf_i386 -o lab6-3 lab6-3.o
tagir@guseinovvb:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab06$ ./lab6-3
Результат: 5
Остаток от деления: 1
tagir@guseinovvb:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab06$
```

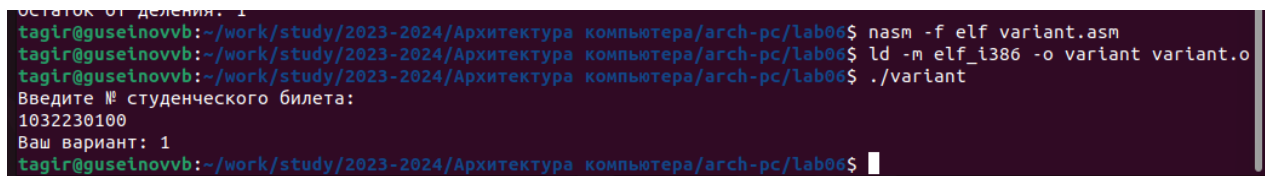
Рис. 15: Работа программы

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



```
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 sprintf
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
```

Рис 16, код Вариант



```
tagir@guseinovvb:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab06$ nasm -f elf variant.asm
tagir@guseinovvb:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab06$ ld -m elf_i386 -o variant variant.o
tagir@guseinovvb:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab06$ ./variant
Введите № студенческого билета:
1032230100
Ваш вариант: 1
tagir@guseinovvb:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab06$
```

Рис. 17 Работа кода вариант.

Ответы на вопросы:

- 1. "mov eax, rem" и "call sprint" в секции кода отвечают за вывод сообщения "Ваш вариант:" на экран.
- ☐ 2. "mov ecx, x" и "mov edx, 80" загружают адрес буфера (x) и длину буфера (80) соответственно в регистры ecx и edx для вызова подпрограммы sread, которая считывает строку из консоли.
- ☐ 3. "call atoi" вызывает подпрограмму atoi для преобразования ASCII кодов символов в число, результат которого сохраняется в регистре eax.
- ☐ 4. Код для вычисления варианта начинается с "xor edx, edx" и "mov ebx, 20", после чего происходит деление числа, сохраненного в eax, на 20 с помощью инструкции "div ebx". Результат деления (цифра варианта) записывается в нижнюю часть регистра AX (AL). Затем происходит увеличение цифры варианта на единицу с помощью инструкции "inc edx".

- 5. Результат деления (цифра варианта) записывается в нижнюю часть регистра AX (AL).
- 6. Инструкция "inc edx" увеличивает цифру варианта на единицу для того, чтобы результат деления не оказывался равным нулю.
- 7. "mov eax, edx" загружает цифру варианта из AL (нижней части регистра AX) в регистр eax для вывода результата на экран с помощью подпрограммы iprintLF.

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. 16, рис. 17)

Получили вариант 1 -

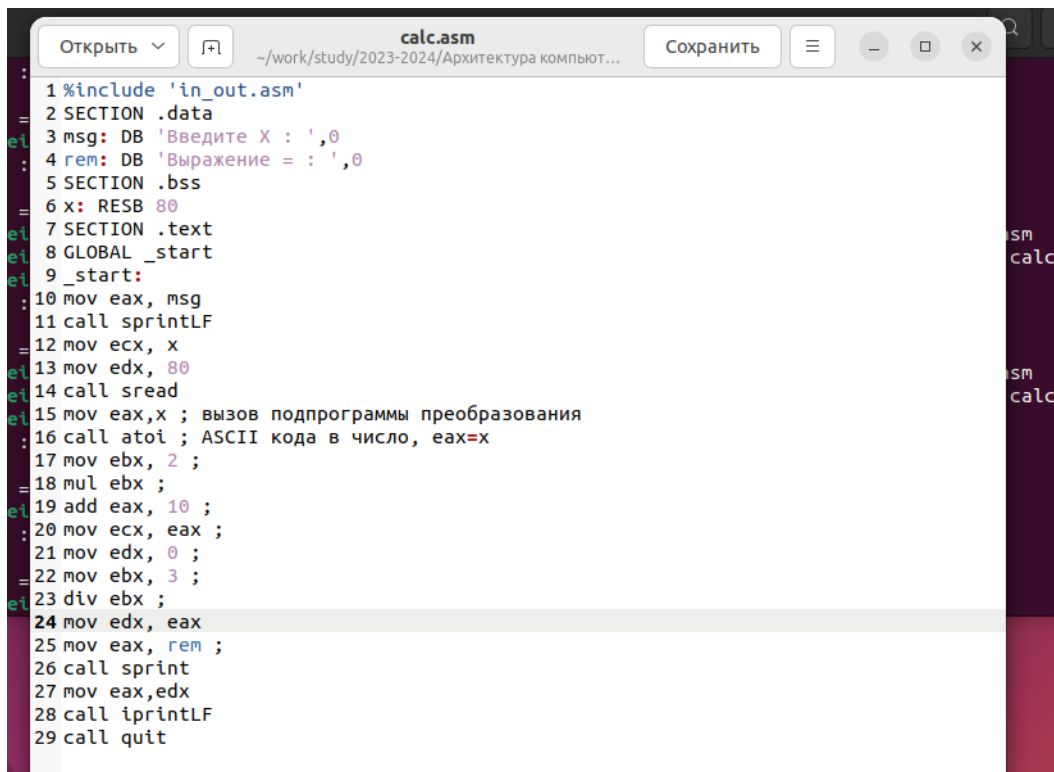
$$(10 + 2x)/3$$

для $x_1 = 1$ и $x_2 = 10$



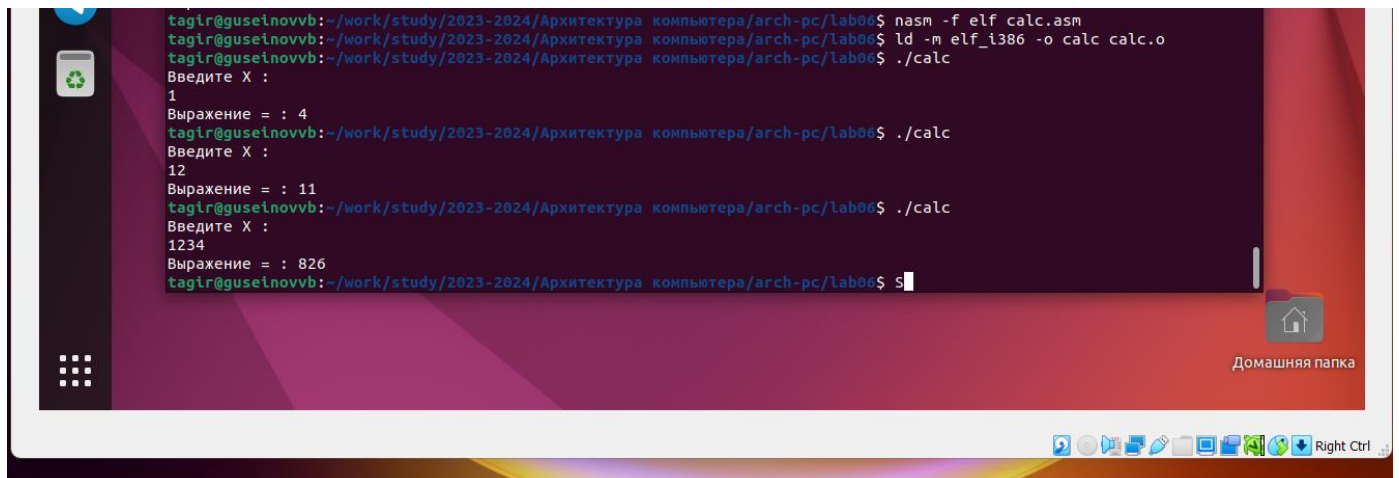
```
tagir@guseinovvb:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab06$ touch calc.asm
tagir@guseinovvb:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab06$
```

Рис 18 Создание файла calc



```
1 %include 'in_out.asm'
2 SECTION .data
3 msg: DB 'Введите X : ',0
4 rem: DB 'Выражение = : ',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 mov ebx, 2 ;
18 mul ebx ;
19 add eax, 10 ;
20 mov ecx, eax ;
21 mov edx, 0 ;
22 mov ebx, 3 ;
23 div ebx ;
24 mov edx, eax
25 mov eax, rem ;
26 call sprint
27 mov eax,edx
28 call iprintLF
29 call quit
```

Рис. 19 Код программы calc



```
tagir@guseinovvb:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab06$ nasm -f elf calc.asm
tagir@guseinovvb:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab06$ ld -m elf_i386 -o calc calc.o
tagir@guseinovvb:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab06$ ./calc
Введите X :
1
Выражение = : 4
tagir@guseinovvb:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab06$ ./calc
Введите X :
12
Выражение = : 11
tagir@guseinovvb:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab06$ ./calc
Введите X :
1234
Выражение = : 826
tagir@guseinovvb:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab06$ s
```

Рис. 20: Работа программы для $(10 + 2x)/3$

как видим все работает исправно программа решает пример для всех представленных значений X.

потом был гит пул но скрина нет.

4 Выводы

В ходе выполнения работы, я освоил работу с арифметическими операциями на языке assembly.