

Отчёта по лабораторной работе 6

Освоение арифметических инструкций языка ассемблера NASM.

Сирота Лев Леонидович НБИбд-04-22

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Выводы	18

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	8
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	13
2.14	Пример программы	14
2.15	Работа программы	14
2.16	Пример программы	16
2.17	Работа программы	17

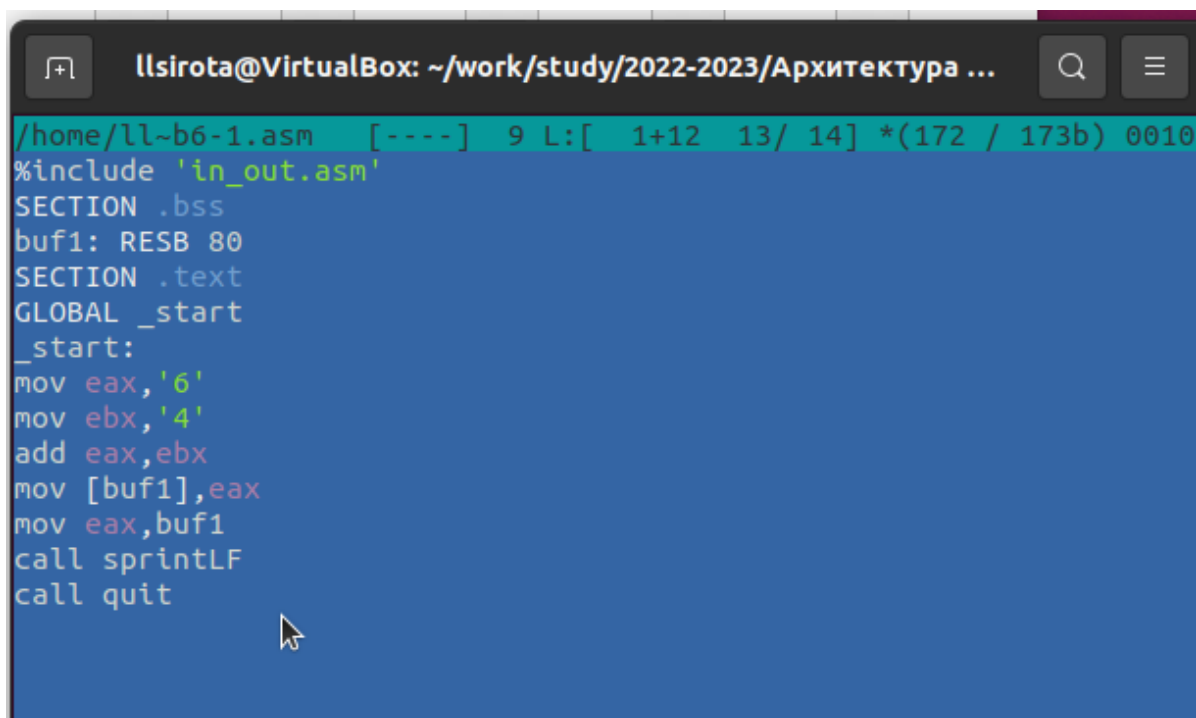
Список таблиц

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])



```
llsirota@VirtualBox: ~/work/study/2022-2023/Архитектура ...
/home/ll~b6-1.asm [----] 9 L:[ 1+12 13/ 14] *(172 / 173b) 0010
#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

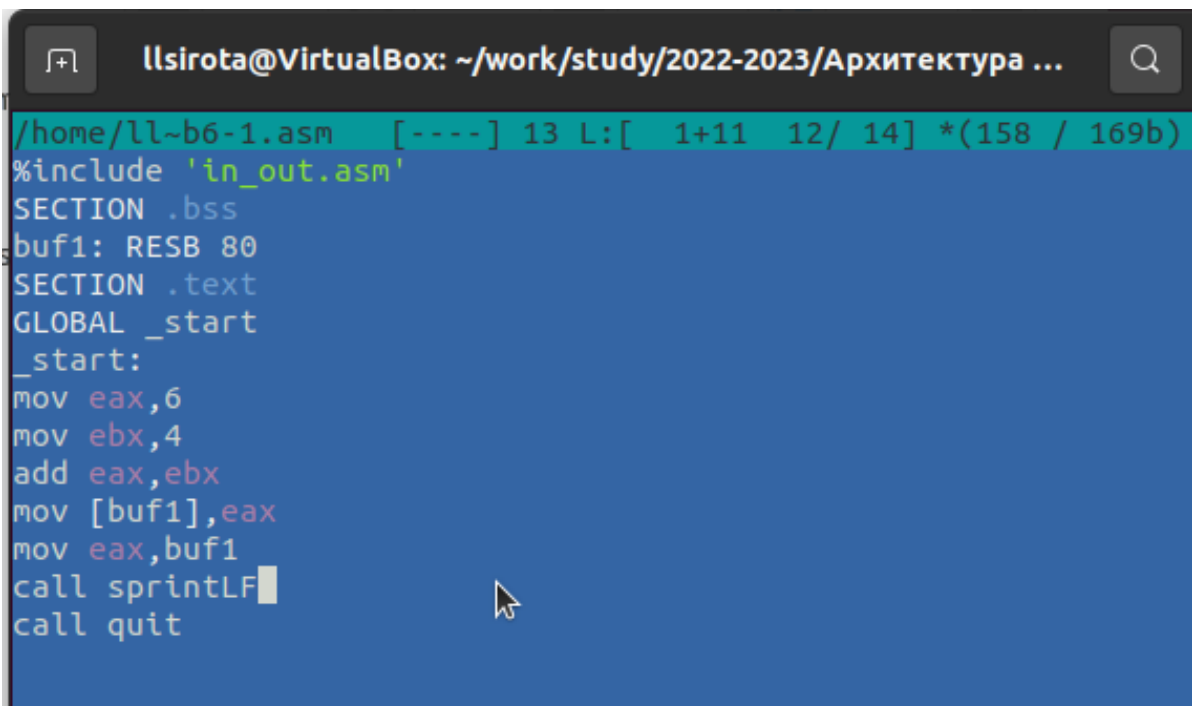
```

llsirota@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/l
ab06$ nasm -f lab6-1.asm
nasm: fatal: unrecognised output format 'lab6-1.asm' - use -hf for a list
type 'nasm -h' for help
llsirota@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/l
ab06$ nasm -f elf lab6-1.asm
llsirota@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/l
ab06$ ld -m elf_i386 -o lab6-1 lab6-1.o
llsirota@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/l
ab06$ ./lab6-1
j
llsirota@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/l
ab06$ █

```

Рис. 2.2: Работа программы

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



```

llsirota@VirtualBox: ~/work/study/2022-2023/Архитектура ...
/home/ll~b6-1.asm [----] 13 L: [ 1+11 12/ 14] *(158 / 169b)
%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

```

llsirota@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/l
ab06$ nasm -f elf lab6-1.asm
llsirota@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/l
ab06$ ld -m elf_i386 -o lab6-1 lab6-1.o
llsirota@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/l
ab06$ ./lab6-1

llsirota@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/l
ab06$

```

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])

```

mc [llsirota@VirtualBox]:~/work/study/2022-2023/Архитект...
/home/ll~b6-2.asm [----] 13 L:[ 1+ 7 8/ 10] *(107 / 118b) 0010 0x0
#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


```
llsirota@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/l  
ab06$ nasm -f elf lab6-2.asm  
llsirota@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/l  
ab06$ ld -m elf_i386 -o lab6-2 lab6-2.o  
llsirota@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/l  
ab06$ ./lab6-2  
106  
llsirota@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/l  
ab06$
```

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 [llsirota@VirtualBox]:~/work/study/2022-2023/Архитект...  
/home/ll~b6-2.asm [----] 0 L:[ 1+ 9 10/ 10] *(114 / 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

```
llsirota@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/l  
ab06$ nasm -f elf lab6-2.asm  
llsirota@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/l  
ab06$ ld -m elf_i386 -o lab6-2 lab6-2.o  
llsirota@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/l  
ab06$ ./lab6-2  
10  
llsirota@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/l  
ab06$
```

Рис. 2.8: 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. [2.9])

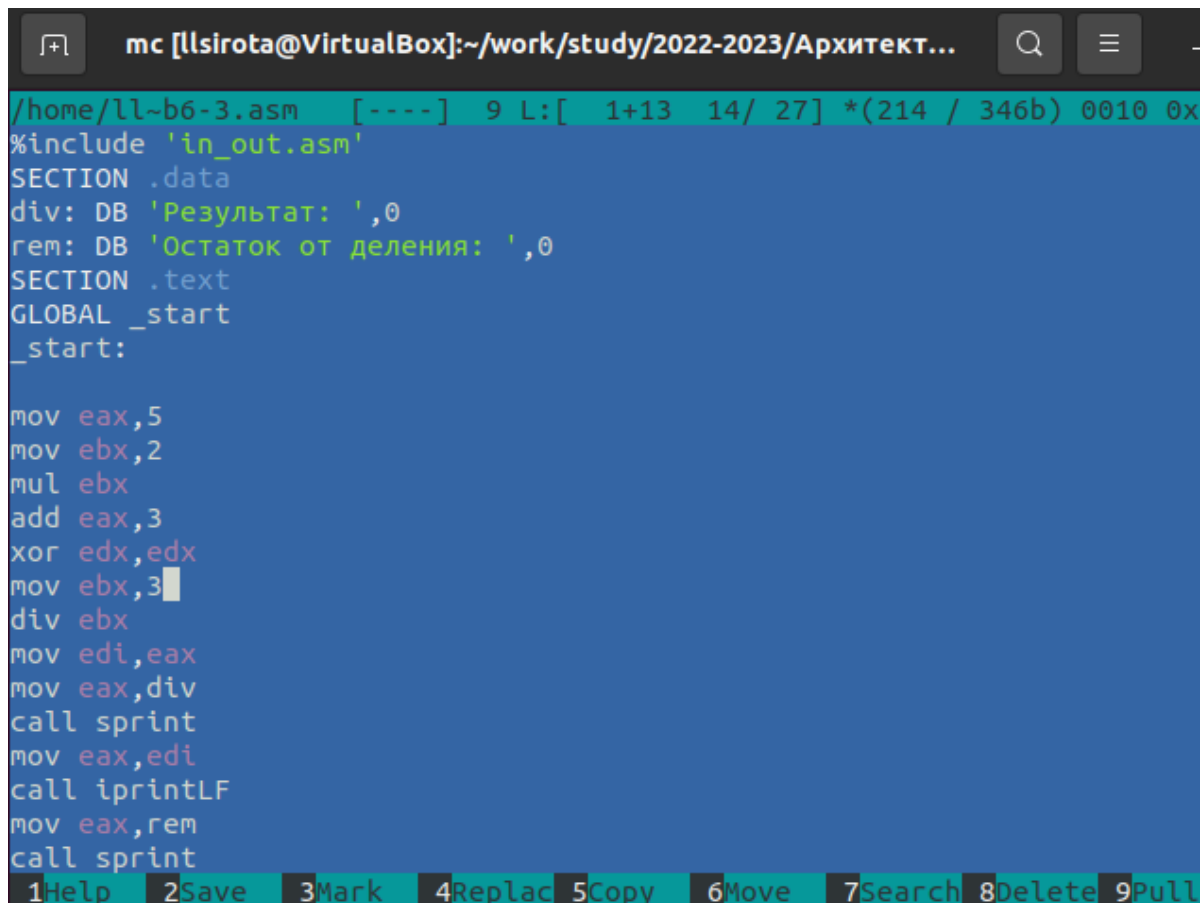
```
llsirota@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/l
ab06$ nasm -f elf lab6-2.asm
llsirota@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/l
ab06$ ld -m elf_i386 -o lab6-2 lab6-2.o
llsirota@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/l
ab06$ ./lab6-2
10llsirota@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs
/l
ab06$
```

Рис. 2.9: Работа программы

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

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

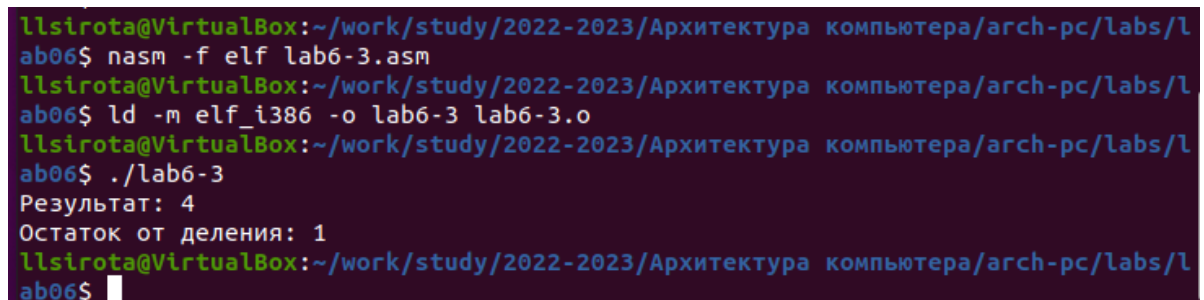
. (рис. [2.10], рис. [2.11])



```
mc [llsirota@VirtualBox]:~/work/study/2022-2023/Архитект...
/home/ll~b6-3.asm [----] 9 L: [ 1+13 14/ 27] *(214 / 346b) 0010 0x
#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
```

Рис. 2.10: Пример программы



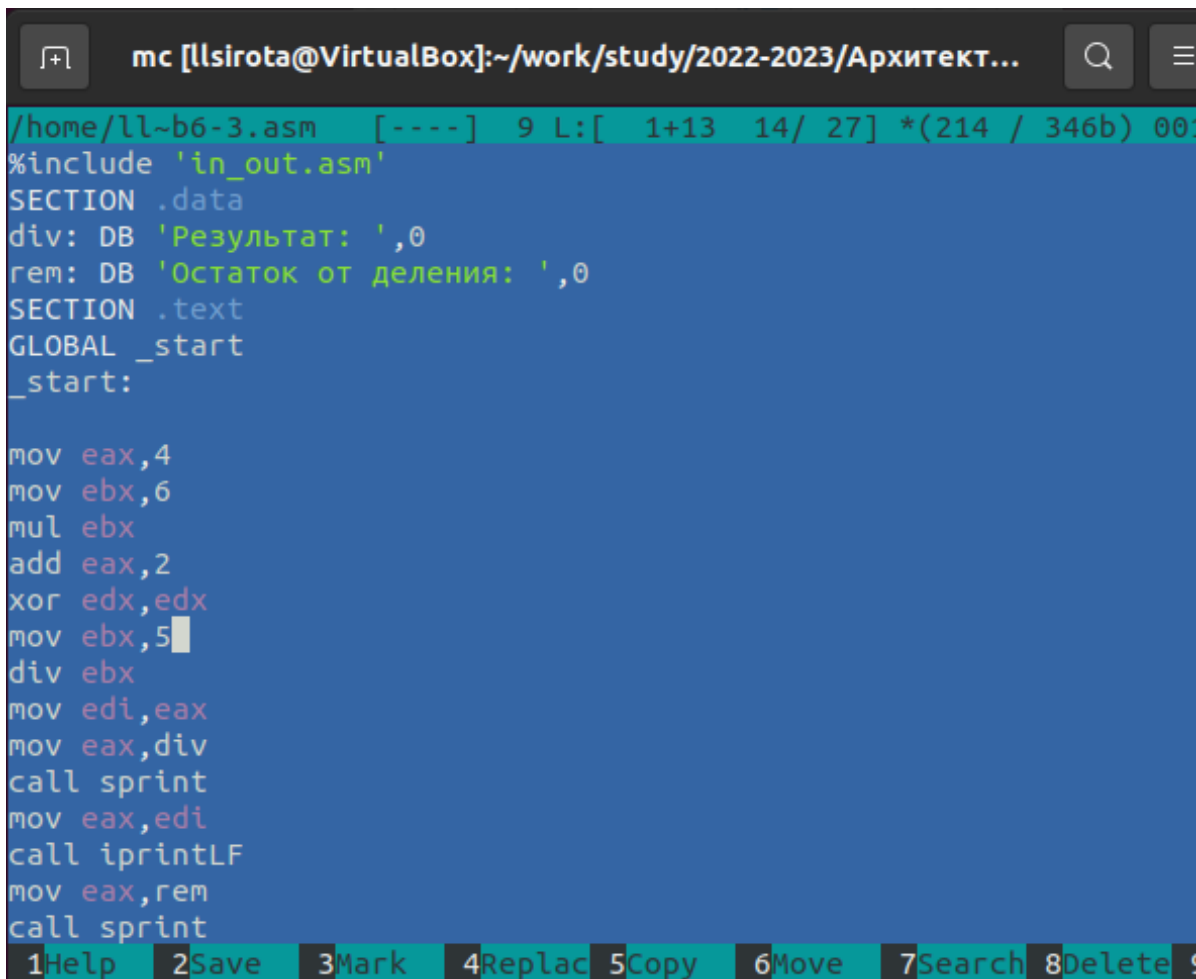
```
llsirota@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/l
ab06$ nasm -f elf lab6-3.asm
llsirota@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/l
ab06$ ld -m elf_i386 -o lab6-3 lab6-3.o
llsirota@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/l
ab06$ ./lab6-3
Результат: 4
Остаток от деления: 1
llsirota@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/l
ab06$
```

Рис. 2.11: Работа программы

Измените текст программы для вычисления выражения

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

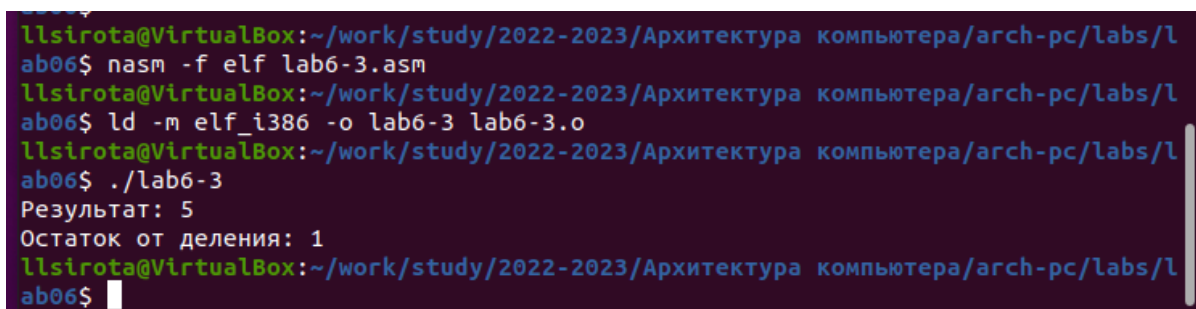
. Создайте исполняемый файл и проверьте его работу. (рис. [2.12], рис. [2.13])



```
/home/ll~b6-3.asm [----] 9 L:[ 1+13 14/ 27] *(214 / 346b) 00:
%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
```

Рис. 2.12: Пример программы

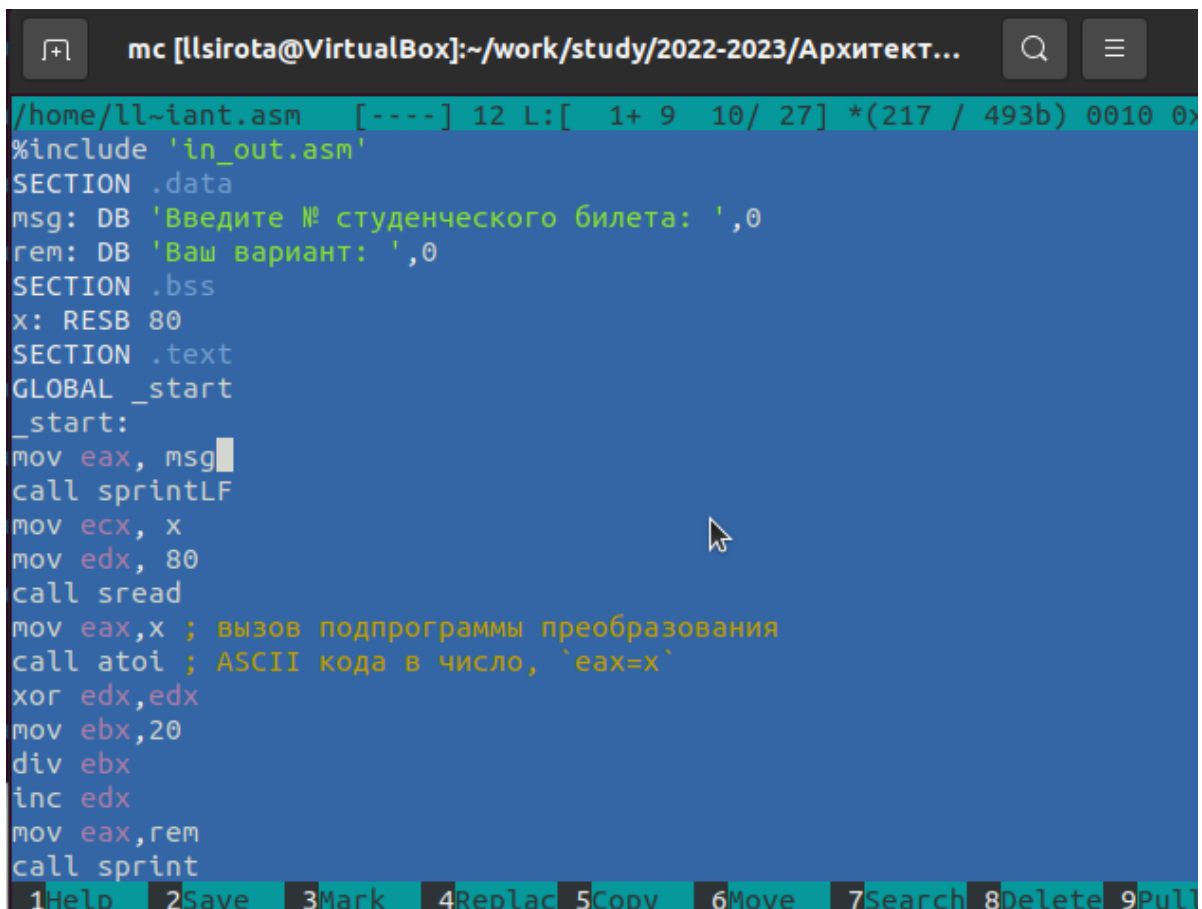


```
llsirota@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/l
ab06$ nasm -f elf lab6-3.asm
llsirota@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/l
ab06$ ld -m elf_i386 -o lab6-3 lab6-3.o
llsirota@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/l
ab06$ ./lab6-3
Результат: 5
Остаток от деления: 1
llsirota@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/l
ab06$
```

Рис. 2.13: Работа программы

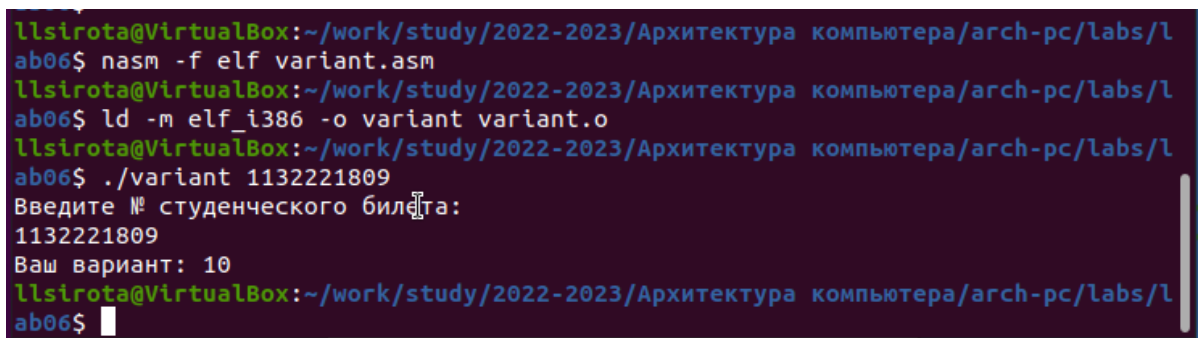
7. В качестве другого примера рассмотрим программу вычисления варианта

задания по номеру студенческого билета, работающую по следующему алгоритму: (рис. [2.14], рис. [2.15])



```
mc [llsirota@VirtualBox]:~/work/study/2022-2023/Архитект...
/home/ll~iant.asm [----] 12 L:[ 1+ 9 10/ 27] *(217 / 493b) 0010 0>
%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 sprint
1Help 2Save 3Mark 4Replac 5Copy 6Move 7Search 8Delete 9Pull
```

Рис. 2.14: Пример программы



```
llsirota@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/l
ab06$ nasm -f elf variant.asm
llsirota@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/l
ab06$ ld -m elf_i386 -o variant variant.o
llsirota@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/l
ab06$ ./variant 1132221809
Введите № студенческого билета:
1132221809
Ваш вариант: 10
llsirota@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/l
ab06$
```

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? `mov ecx, x` `mov edx, 80` `call sread`

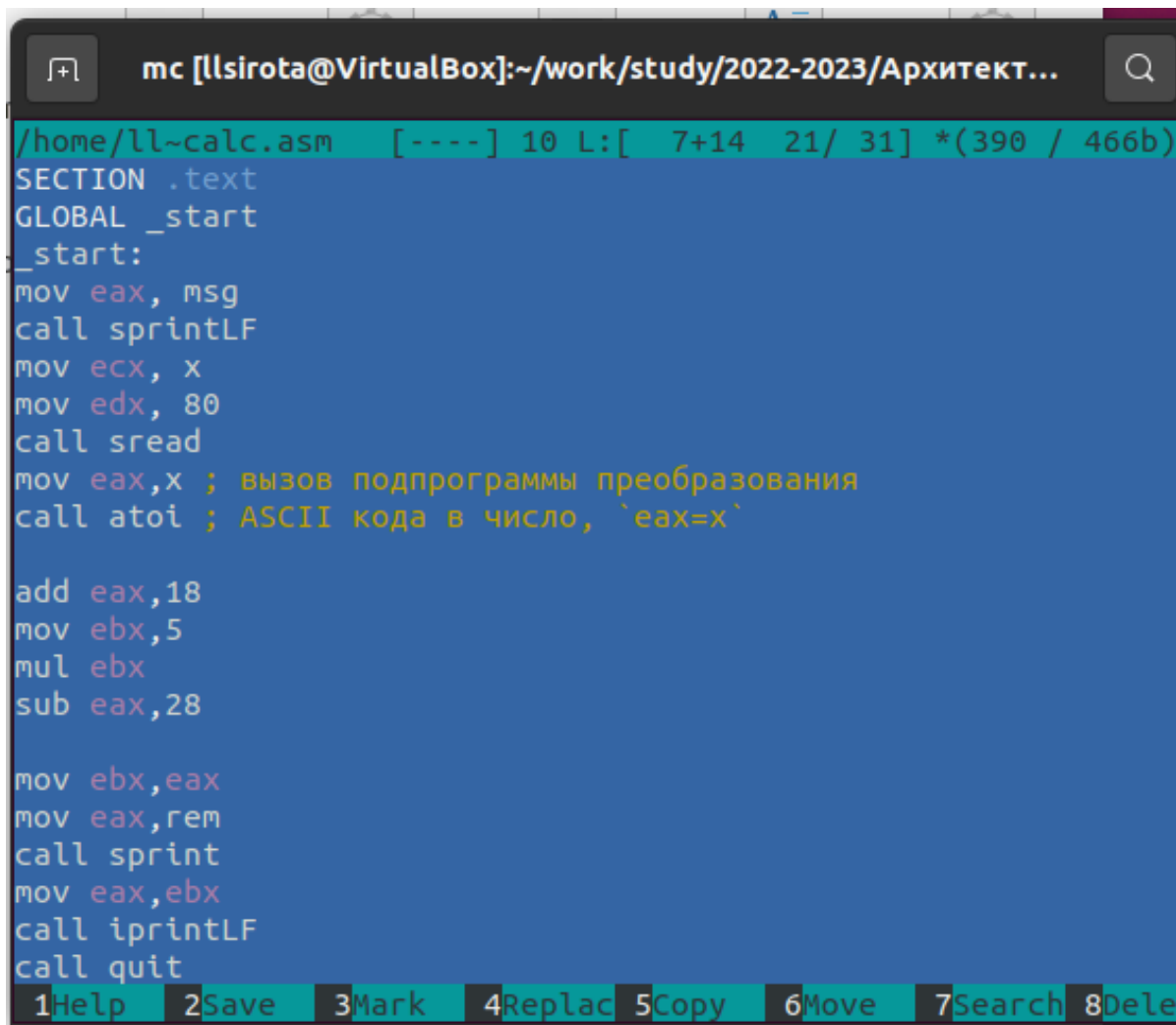
Считывает значение студбилета в переменную X из консоли

- 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 10 -

$$5(x + 18) - 28$$

для $x=2$ и 3



```
mc [llsirota@VirtualBox]:~/work/study/2022-2023/Архитект...  
/home/ll~calc.asm [----] 10 L:[ 7+14 21/ 31] *(390 / 466b)  
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`  
  
add eax, 18  
mov ebx, 5  
mul ebx  
sub eax, 28  
  
mov ebx, eax  
mov eax, rem  
call sprintf  
mov eax, ebx  
call iprintLF  
call quit  
1Help 2Save 3Mark 4Replac 5Copy 6Move 7Search 8Dele
```

Рис. 2.16: Пример программы


```
llsirota@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/l
ab06$ nasm -f elf calc.asm
llsirota@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/l
ab06$ ld -m elf_i386 -o calc calc.o
llsirota@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/l
ab06$ ./calc
Введите X
2
выражение = : 72
llsirota@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/l
ab06$ ./calc
Введите X
3
выражение = : 77
llsirota@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/l
ab06$
llsirota@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/l
ab06$
```

Рис. 2.17: Работа программы

3 Выводы

Изучил работу с арифметическими операциями