

# **Лабораторная работа №6**

Седохин Даниил Алексеевич

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>4</b>
<b>2</b>	<b>Выполнение лабораторной работы</b>	<b>5</b>
2.1	Ответы на вопросы: . . . . .	17
<b>3</b>	<b>Задание для самостоятельной работы</b>	<b>18</b>
<b>4</b>	<b>Выводы</b>	<b>19</b>

## Список иллюстраций

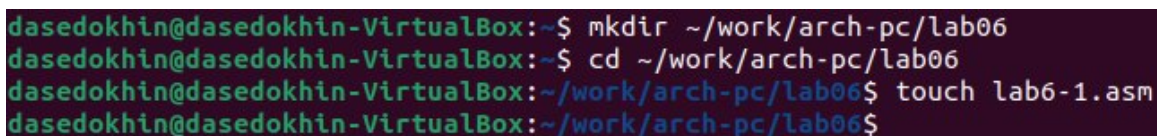
2.1	Создание каталога для лабораторной работы №6 и файла lab6-1.asm	5
2.2	Ввод в файл lab6-1.asm текст программы из листинга 6.1 . . . . .	6
2.3	Создание и проверка исполняемого файла . . . . .	6
2.4	Редактирование текста программы из листинга 6.1 . . . . .	7
2.5	Создание и проверка исполняемого файла . . . . .	7
2.6	Таблица ASCII . . . . .	7
2.7	Создание файла lab6-2.asm . . . . .	8
2.8	Редактирование файла lab6-2.asm из листинга 6.2 . . . . .	8
2.9	Создание и проверка исполняемого файла . . . . .	9
2.10	Редактирование файла lab6-2.asm . . . . .	9
2.11	Создание и проверка исполняемого файла . . . . .	10
2.12	Замена функции iprintLF на iprint . . . . .	10
2.13	Создание и проверка файла исполняемого файла . . . . .	10
2.14	Создание файла lab6-3.asm . . . . .	11
2.15	Ввод листинга 6.3 в файл lab6-3.asm . . . . .	12
2.16	Создание и проверка исполняемого файла . . . . .	13
2.17	Редактирование текста программы lab6-3.asm . . . . .	14
2.18	Создание и проверка исполняемого файла . . . . .	15
2.19	Создание файла variant.asm . . . . .	15
2.20	Ввод листинга 6.4 в файл variant.asm . . . . .	16
2.21	Создание и проверка исполняемого файла . . . . .	17
3.1	Загрузка файлов на github . . . . .	18

# 1 Цель работы

Освоение арифметических инструкций языка ассемблера NASM.

## 2 Выполнение лабораторной работы

- 1) Создаём каталог для программ лабораторной работы № 6, перейдём в него и создадим файл lab6-1.asm (рис. 2.1).

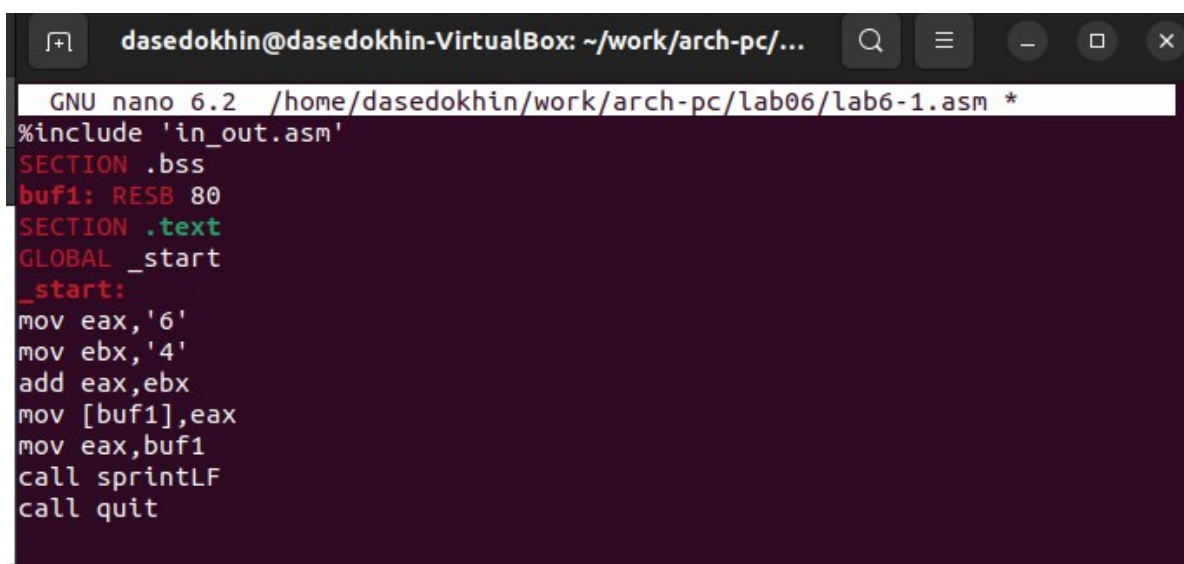


```
dasedokhin@dasedokhin-VirtualBox:~$ mkdir ~/work/arch-pc/lab06
dasedokhin@dasedokhin-VirtualBox:~$ cd ~/work/arch-pc/lab06
dasedokhin@dasedokhin-VirtualBox:~/work/arch-pc/lab06$ touch lab6-1.asm
dasedokhin@dasedokhin-VirtualBox:~/work/arch-pc/lab06$
```

Рис. 2.1: Создание каталога для лабораторной работы №6 и файла lab6-1.asm

- 2) Рассмотрим примеры программ вывода символьных и численных значений. Программы будут выводить значения записанные в регистр еах. Введём в файл lab6-1.asm текст программы из листинга 6.1. (рис. 2.2).

В данной программе в регистр еах записывается символ 6 (mov еах, '6'), в регистр ебх символ 4 (mov ебх, '4'). Далее к значению в регистре еах прибавляем значение регистра ебх (add еах, ебх, результат сложения запишется в регистр еах). Далее выводим результат. Так как для работы функции sprintLF в регистр еах должен быть записан адрес, необходимо использовать дополнительную переменную. Для этого запишем значение регистра еах в переменную buf1 (mov [buf1], еах), а затем запишем адрес переменной buf1 в регистр еах (mov еах, buf1) и вызовем функцию sprintLF.

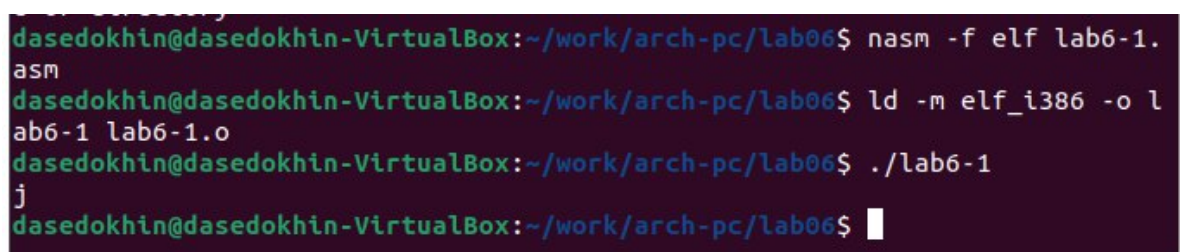


```
GNU nano 6.2 /home/dasedokhin/work/arch-pc/lab06/lab6-1.asm *
#include 'in_out.asm'
SECTION .bss
buf1: RESB 80
SECTION .text
GLOBAL _start
_start:
mov eax, '6'
mov ebx, '4'
add eax, ebx
mov [buf1], eax
mov eax, buf1
call sprintf
call quit
```

Рис. 2.2: Ввод в файл lab6-1.asm текст программы из листинга 6.1

- 3) Создадим исполняемый файл и проверим его. (рис. 2.3).

Пояснение: В данном случае при выводе значения регистра `eax` мы ожидаем увидеть число 10. Однако результатом будет символ `j`. Это происходит потому, что код символа 6 равен 00110110 в двоичном представлении (или 54 в десятичном представлении), а код символа 4 – 00110100 (52). Команда `add eax, ebx` запишет в регистр `eax` сумму кодов – 01101010 (106), что в свою очередь является кодом символа `j`.



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

Рис. 2.3: Создание и проверка исполняемого файла

- 4) Далее изменим текст программы и вместо символов, запишем в регистры числа. Исправим текст программы (Листинг 6.1). (рис. 2.4).

```
dasedokhin@dasedokhin-VirtualBox: ~/work/arch-pc/...
GNU nano 6.2 /home/dasedokhin/work/arch-pc/lab06/lab6-1.asm *
%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 sprintf
call quit
```

Рис. 2.4: Редактирование текста программы из листинга 6.1

Создадим исполняемый файл и запустим его. (рис. 2.5).

```
dasedokhin@dasedokhin-VirtualBox:~/work/arch-pc/lab06$ nasm -f elf lab6-1.asm
dasedokhin@dasedokhin-VirtualBox:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-1 lab6-1.o
dasedokhin@dasedokhin-VirtualBox:~/work/arch-pc/lab06$ ./lab6-1
dasedokhin@dasedokhin-VirtualBox:~/work/arch-pc/lab06$
```

Рис. 2.5: Создание и проверка исполняемого файла

Как и в предыдущем случае при исполнении программы мы не получим число 10. В данном случае выводится символ с кодом 10. Пользуясь таблицей ASCII определим какому символу соответствует код 10. (рис. 2.6).

Данный символ не отображается на экране.

9	9	1001	11	[HORIZONTAL TAB]
10	A	1010	12	[LINE FEED]
11	B	1011	13	[VERTICAL TAB]

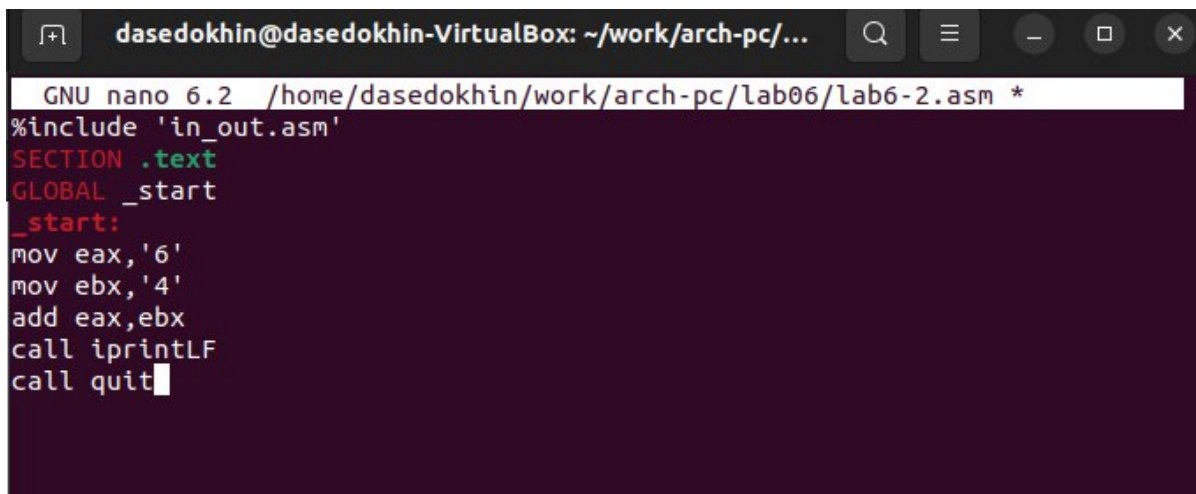
Рис. 2.6: Таблица ASCII

- 5) 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 6.1 с использованием этих функций. Создадим файл `lab6-2.asm` в каталоге `~/work/arch-pc/lab06` . (рис. 2.7).

```
dasedokhin@dasedokhin-VirtualBox:~/work/arch-pc/lab06$ touch ~/work/arch-pc/lab06/lab6-2.asm
dasedokhin@dasedokhin-VirtualBox:~/work/arch-pc/lab06$ ls
in_out.asm  lab6-1  lab6-1.asm  lab6-1.o  lab6-2.asm
dasedokhin@dasedokhin-VirtualBox:~/work/arch-pc/lab06$
```

Рис. 2.7: Создание файла `lab6-2.asm`

Введём в него текст программы из листинга 6.2 (рис. 2.8).



```
GNU nano 6.2 /home/dasedokhin/work/arch-pc/lab06/lab6-2.asm *
%include 'in_out.asm'
SECTION .text
GLOBAL _start
_start:
mov eax, '6'
mov ebx, '4'
add eax, ebx
call iprintLF
call quit
```

Рис. 2.8: Редактирование файла `lab6-2.asm` из листинга 6.2

Создадим исполняемый файл и запустим его. (рис. 2.9)

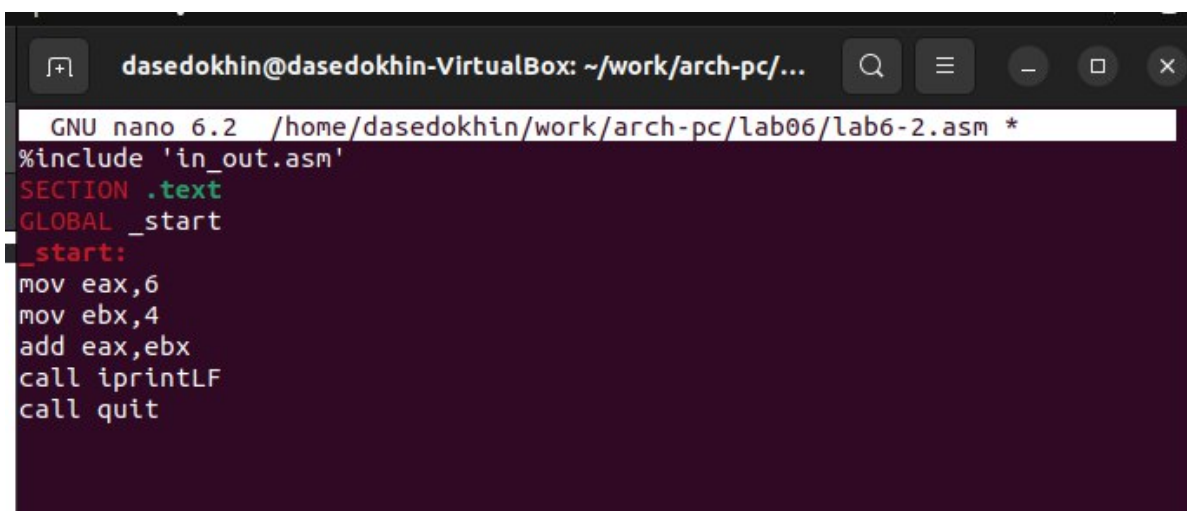


```
dasedokhin@dasedokhin-VirtualBox:~/work/arch-pc/lab06$ nasm -f elf lab6-2.asm
dasedokhin@dasedokhin-VirtualBox:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-2 lab6-2.o
dasedokhin@dasedokhin-VirtualBox:~/work/arch-pc/lab06$ ./lab6-2
106
dasedokhin@dasedokhin-VirtualBox:~/work/arch-pc/lab06$
```

Рис. 2.9: Создание и проверка исполняемого файла

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 6.1, функция `iprintLF` позволяет вывести число, а не символ, кодом которого является это число.

- 6) Аналогично предыдущему примеру изменим символы на числа, заменив строки. (рис. 2.10).



```
GNU nano 6.2 /home/dasedokhin/work/arch-pc/lab06/lab6-2.asm *
%include 'in_out.asm'
SECTION .text
GLOBAL _start
_start:
mov eax,6
mov ebx,4
add eax,ebx
call iprintLF
call quit
```

Рис. 2.10: Редактирование файла lab6-2.asm

Создадим исполняемый файл и запустим его. (рис. 2.11).

```

dasedokhin@dasedokhin-VirtualBox:~/work/arch-pc/lab06$ nasm -f elf lab6-2.asm
dasedokhin@dasedokhin-VirtualBox:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-2 lab6-2.o
dasedokhin@dasedokhin-VirtualBox:~/work/arch-pc/lab06$ ./lab6-2
10
dasedokhin@dasedokhin-VirtualBox:~/work/arch-pc/lab06$

```

Рис. 2.11: Создание и проверка исполняемого файла

В результате при запуске исполняемого файла мы получаем число 10. Заменяем функцию `iprintLF` на `iprint`. (рис. 2.12).

```

GNU nano 6.2 /home/dasedokhin/work/arch-pc/lab06/lab6-2.asm *
#include 'in_out.asm'
SECTION .text
GLOBAL _start
_start:
mov eax,6
mov ebx,4
add eax,ebx
call iprint
call quit

```

Рис. 2.12: Замена функции `iprintLF` на `iprint`

Создадим исполняемый файл и запустим его. (рис. 2.13).

```

dasedokhin@dasedokhin-VirtualBox:~/work/arch-pc/lab06$ nasm -f elf lab6-2.asm
dasedokhin@dasedokhin-VirtualBox:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-2 lab6-2.o
dasedokhin@dasedokhin-VirtualBox:~/work/arch-pc/lab06$ ./lab6-2
10dasedokhin@dasedokhin-VirtualBox:~/work/arch-pc/lab06$

```

Рис. 2.13: Создание и проверка файла исполняемого файла

Вывод функции `iprintLF` отличается от функции `iprint` тем, что выводимое число не записывается на отдельной строке при использовании функции `iprint`.

- 7) В качестве примера выполнения арифметических операций в NASM приведем программу вычисления арифметического выражения  $\text{X}(\text{X}) = (5 \times 2 + 3)/3$ . Создадим файл lab6-3.asm в каталоге ~/work/arch-pc/lab06 (рис. 2.14).

```
dasedokhin@dasedokhin-VirtualBox:~$ cd ~/work/arch-pc/lab06
dasedokhin@dasedokhin-VirtualBox:~/work/arch-pc/lab06$ touch ~/work/arch-pc/lab06/lab6-3.asm
dasedokhin@dasedokhin-VirtualBox:~/work/arch-pc/lab06$ ls
in_out.asm  lab6-1.asm  lab6-2      lab6-2.o
lab6-1      lab6-1.o    lab6-2.asm  lab6-3.asm
dasedokhin@dasedokhin-VirtualBox:~/work/arch-pc/lab06$
```

Рис. 2.14: Создание файла lab6-3.asm

- 8) Введём в lab6-3.asm текст из листинга 6.3 (рис. 2.15).

```
dasedokhin@dasedokhin-VirtualBox: ~/work/arch-pc/...
GNU nano 6.2 /home/dasedokhin/work/arch-pc/lab06/lab6-3.asm *
;-----
; Программа вычисления выражения
;-----
#include 'in_out.asm' ; подключение внешнего файла
SECTION .data
div: DB 'Результат: ',0
rem: DB 'Остаток от деления: ',0
SECTION .text
GLOBAL _start
_start:
; ---- Вычисление выражения
mov eax,5 ; EAX=5
mov ebx,2 ; EBX=2
mul ebx ; EAX=EAX*EBX
add eax,3 ; EAX=EAX+3
xor edx,edx ; обнуляем EDX для корректной работы div
mov ebx,3 ; EBX=3
div ebx ; EAX=EAX/3, EDX=остаток от деления
mov edi,eax ; запись результата вычисления в 'edi'
; ---- Вывод результата на экран
mov eax,div ; вызов подпрограммы печати
call sprint ; сообщения 'Результат: '
mov eax,edi ; вызов подпрограммы печати значения
call iprintLF ; из 'edi' в виде символов
mov eax,rem ; вызов подпрограммы печати
call sprint ; сообщения 'Остаток от деления: '
mov eax,edx ; вызов подпрограммы печати значения
call iprintLF ; из 'edx' (остаток) в виде символов
call quit ; вызов подпрограммы завершения
```

Рис. 2.15: Ввод листинга 6.3 в файл lab6-3.asm

9) Создадим исполняемый файл и запустим его. (рис. 2.16).

```

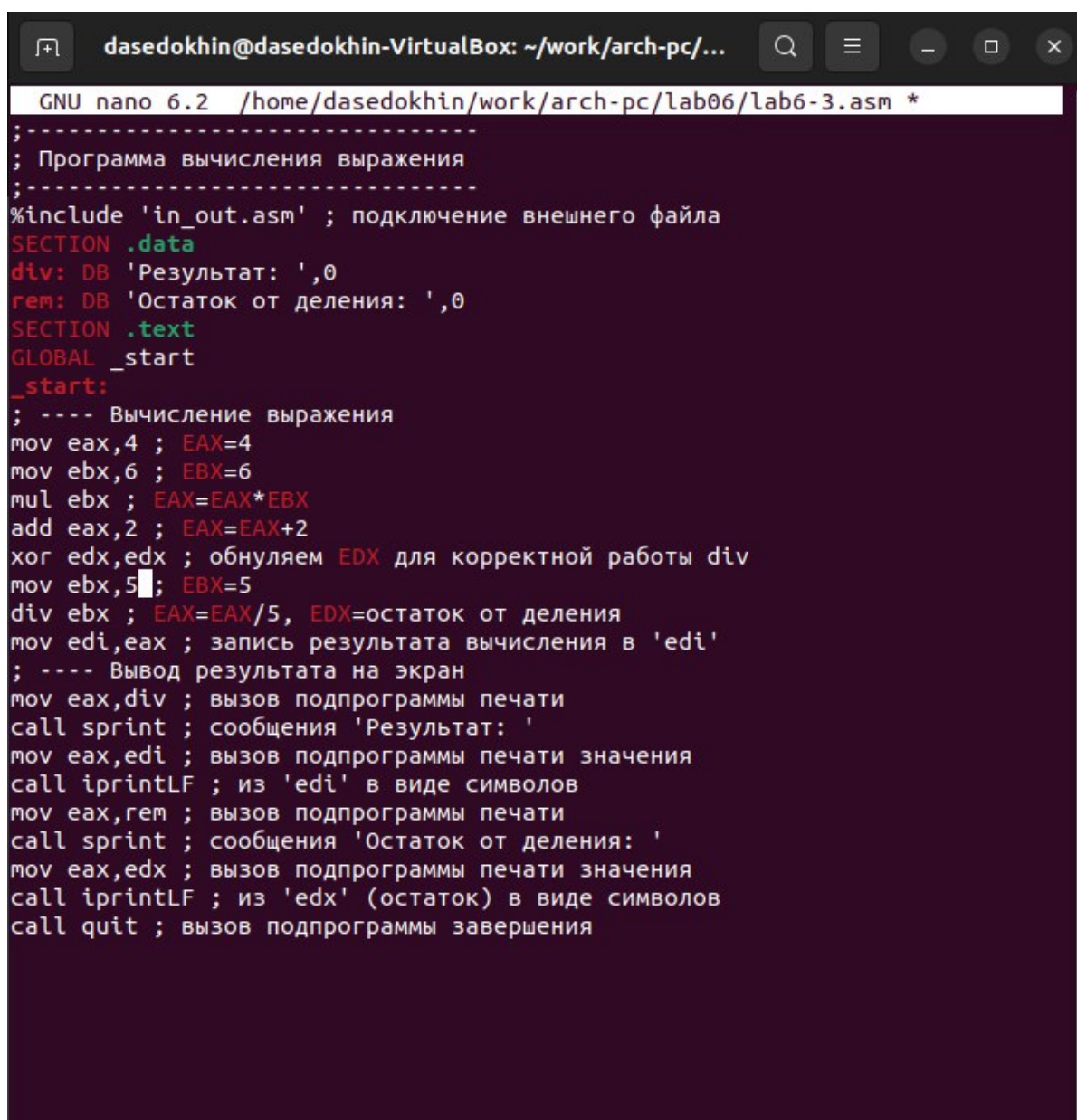
dasedokhin@dasedokhin-VirtualBox:~/work/arch-pc/lab06$ nasm -f elf lab6-3.
asm
dasedokhin@dasedokhin-VirtualBox:~/work/arch-pc/lab06$ ld -m elf_i386 -o l
ab6-3 lab6-3.o
dasedokhin@dasedokhin-VirtualBox:~/work/arch-pc/lab06$ ./lab6-3
Результат: 4
Остаток от деления: 1
dasedokhin@dasedokhin-VirtualBox:~/work/arch-pc/lab06$

```

Рис. 2.16: Создание и проверка исполняемого файла

- 10) Изменим текст программы для вычисления выражения  $\text{div}(x) = (4 \times 6 + 2)/5$ .  
(рис. 2.17).





```
GNU nano 6.2 /home/dasedokhin/work/arch-pc/lab06/lab6-3.asm *
;-----
; Программа вычисления выражения
;-----
#include 'in_out.asm' ; подключение внешнего файла
SECTION .data
div: DB 'Результат: ',0
rem: DB 'Остаток от деления: ',0
SECTION .text
GLOBAL _start
_start:
; ---- Вычисление выражения
mov eax,4 ; EAX=4
mov ebx,6 ; EBX=6
mul ebx ; EAX=EAX*EBX
add eax,2 ; EAX=EAX+2
xor edx,edx ; обнуляем EDX для корректной работы div
mov ebx,5 ; EBX=5
div ebx ; EAX=EAX/5, EDX=остаток от деления
mov edi,eax ; запись результата вычисления в 'edi'
; ---- Вывод результата на экран
mov eax,div ; вызов подпрограммы печати
call sprint ; сообщения 'Результат: '
mov eax,edi ; вызов подпрограммы печати значения
call iprintLF ; из 'edi' в виде символов
mov eax,rem ; вызов подпрограммы печати
call sprint ; сообщения 'Остаток от деления: '
mov eax,edx ; вызов подпрограммы печати значения
call iprintLF ; из 'edx' (остаток) в виде символов
call quit ; вызов подпрограммы завершения
```

Рис. 2.17: Редактирование текста программы lab6-3.asm

Создадим исполняемый файл и проверим его работу. (рис. 2.18).

```
dasedokhin@dasedokhin-VirtualBox:~/work/arch-pc/lab06$ nasm -f elf lab6-3.asm
dasedokhin@dasedokhin-VirtualBox:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-3 lab6-3.o
dasedokhin@dasedokhin-VirtualBox:~/work/arch-pc/lab06$ ./lab6-3
Результат: 5
Остаток от деления: 1
dasedokhin@dasedokhin-VirtualBox:~/work/arch-pc/lab06$
```

Рис. 2.18: Создание и проверка исполняемого файла

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

- вывести запрос на введение  $N^{\circ}$  студенческого билета
- вычислить номер варианта по формуле:  $(XX \bmod 20) + 1$ , где  $XX$  – номер студенческого билета (В данном случае  $X \bmod X$  – это остаток от деления  $X$  на  $X$ ).
- вывести на экран номер варианта.

В данном случае число, над которым необходимо проводить арифметические операции, вводится с клавиатуры. Как отмечалось выше ввод с клавиатуры осуществляется в символьном виде и для корректной работы арифметических операций в NASM символы необходимо преобразовать в числа. Для этого может быть использована функция `atoi` из файла `in_out.asm`. Создадим файл `variant.asm` в каталоге `~/work/arch-pc/lab06` (рис. 2.19).

```
dasedokhin@dasedokhin-VirtualBox:~/work/arch-pc/lab06$ touch ~/work/arch-pc/lab06/variant.asm
dasedokhin@dasedokhin-VirtualBox:~/work/arch-pc/lab06$ ls
in_out.asm lab6-1.asm lab6-2 lab6-2.o lab6-3.asm variant.asm
lab6-1 lab6-1.o lab6-2.asm lab6-3 lab6-3.o
dasedokhin@dasedokhin-VirtualBox:~/work/arch-pc/lab06$
```

Рис. 2.19: Создание файла `variant.asm`

Введём в файл `variant.asm` текст из листинга 6.4 (рис. 2.20).

```
GNU nano 6.2 /home/dasedokhin/work/arch-pc/lab06/variant.asm *
;-----
; Программа вычисления варианта
;-----
#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. 2.20: Ввод листинга 6.4 в файл variant.asm

Создадим исполняемый файл и проверим его. (рис. 2.21).



```

dasedokhin@dasedokhin-VirtualBox:~/work/arch-pc/lab06$ nasm -f elf variant
.asm
dasedokhin@dasedokhin-VirtualBox:~/work/arch-pc/lab06$ ld -m elf_i386 -o v
ariant variant.o
dasedokhin@dasedokhin-VirtualBox:~/work/arch-pc/lab06$ ./variant
Введите № студенческого билета:
1132231845
Ваш вариант: 6
dasedokhin@dasedokhin-VirtualBox:~/work/arch-pc/lab06$

```

Рис. 2.21: Создание и проверка исполняемого файла

При аналитическом вычислении номера варианта результат совпал с результатом работы программы.

## 2.1 Ответы на вопросы:

- 1) За вывод строки *Ваш вариант:* отвечают 2 строки: **mov eax,rem call sprint**
- 2) Объявляется переменная, под неё выделяется 80 байт, а затем вызывается функция которая отвечает за ввод строки пользователем в ранее объявленную переменную.
- 3) Это функция преобразования ascii-код символа в целое число
- 4) За вычисление варианта отвечает следующая последовательность строк - **xor edx,edx mov ebx,20 div ebx inc edx**
- 5) Запись происходит в регистр edx.
- 6) Данная инструкция преобразует 1 к значению регистра edx.
- 7) Строки - **mov eax,edx call iprintLF**

отвечают за вывод результата вычислений в консоль

### **3 Задание для самостоятельной работы**

Загрузим файлы на github. (рис. 3.1).

Загрузка файлов на github

Рис. 3.1: Загрузка файлов на github

## 4 Выводы

Я освоил арифметические инструкции языка ассемблера NASM.