

# **Шаблон отчёта по лабораторной работе**

**6**

БЕМБО ЛУМИНГУ ЖОЗЕ

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Задание</b>	<b>6</b>
<b>3</b>	<b>Выполнение лабораторной работы</b>	<b>7</b>
3.1	Ответы на вопросы по программе . . . . .	15
<b>4</b>	<b>Выводы</b>	<b>17</b>
	<b>Список литературы</b>	<b>18</b>

# Список иллюстраций

3.1 создание директории . . . . .	7
3.2 создание файла . . . . .	7
3.3 создание копии файла . . . . .	7
3.4 редактирование файла . . . . .	8
3.5 запуск исполняемого файла . . . . .	8
3.6 редактирование файла . . . . .	9
3.7 запуск исполняемого файла . . . . .	9
3.8 создание копии файла . . . . .	9
3.9 редактирование файла . . . . .	10
3.10 запуск исполняемого файла . . . . .	10
3.11 редактирование файла . . . . .	11
3.12 запуск исполняемого файла . . . . .	12
3.13 запуск исполняемого файла . . . . .	12
3.14 редактирование файла . . . . .	13
3.15 запуск исполняемого файла . . . . .	13
3.16 изменение программы . . . . .	14
3.17 запуск исполняемого файла . . . . .	14
3.18 создание копии файла . . . . .	14
3.19 редактирование файла . . . . .	15
3.20 запуск исполняемого файла . . . . .	15
3.21 запуск исполняемого файла . . . . .	17

## **Список таблиц**

# 1 Цель работы

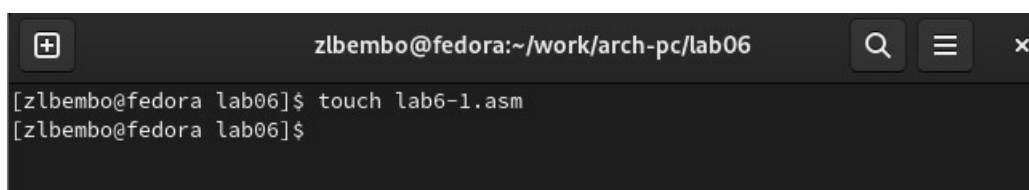
- Цель данной лабораторной работы - освоение арифметических инструкций языка ассемблера NASM..

## 2 Задание

- 1) Символьные и численные данные в NASM
- 2) Выполнение арифметических операций в NASM
- 3) Выполнение заданий для самостоятельной работы

## 3 Выполнение лабораторной работы

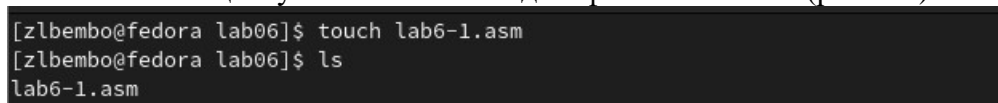
- 1) Символьные и численные данные в NASM
- С помощью утилиты `mkdir` создаю директорию, в которой буду создавать файлы с программами для лабораторной работы №6 (рис. [3.1]). Перехожу в созданный каталог с помощью утилиты `cd` (рис. 3.1).



```
zlbembo@fedora:~/work/arch-pc/lab06
[zlbembo@fedora lab06]$ touch lab6-1.asm
[zlbembo@fedora lab06]$
```

Рис. 3.1: создание директории

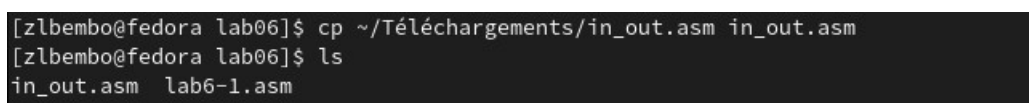
- С помощью утилиты `touch` создаю файл `lab6-1.asm` (рис. 3.2).



```
[zlbembo@fedora lab06]$ touch lab6-1.asm
[zlbembo@fedora lab06]$ ls
lab6-1.asm
```

Рис. 3.2: создание файла

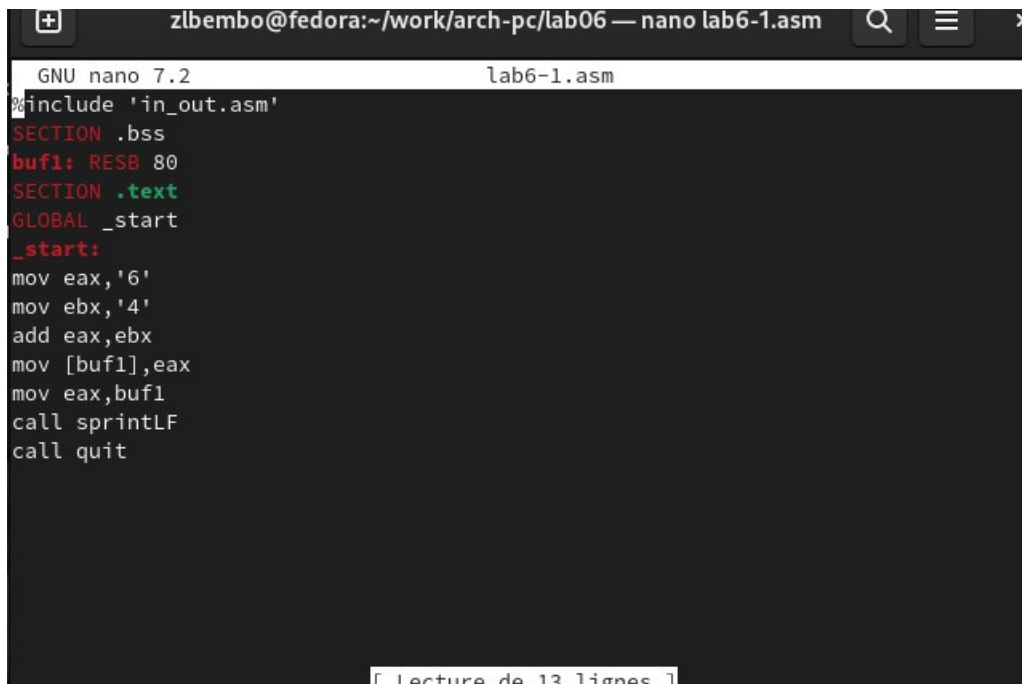
- Копирую в текущий каталог файл `in_out.asm` с помощью утилиты `cp`, т.к. он будет использоваться в других программах (рис. 3.3).



```
[zlbembo@fedora lab06]$ cp ~/Téléchargements/in_out.asm in_out.asm
[zlbembo@fedora lab06]$ ls
in_out.asm lab6-1.asm
```

Рис. 3.3: создание копии файла

- Открываю созданный файл lab6-1.asm, вставляю в него программу вывода значения регистра eax (рис. 3.4).



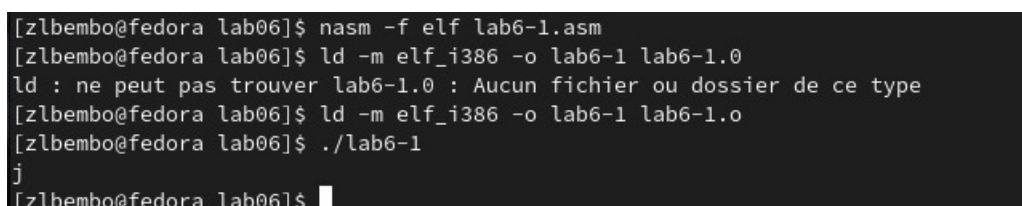
```

GNU nano 7.2 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 printf
call quit

```

Рис. 3.4: редактирование файла

- Создаю исполняемый файл программы и запускаю его (рис. [3.5]). Вывод программы: символ j, потому что программа вывела символ, соответствующий по системе ASCII сумме двоичных кодов символов 4 и 6. (рис. 3.5).



```

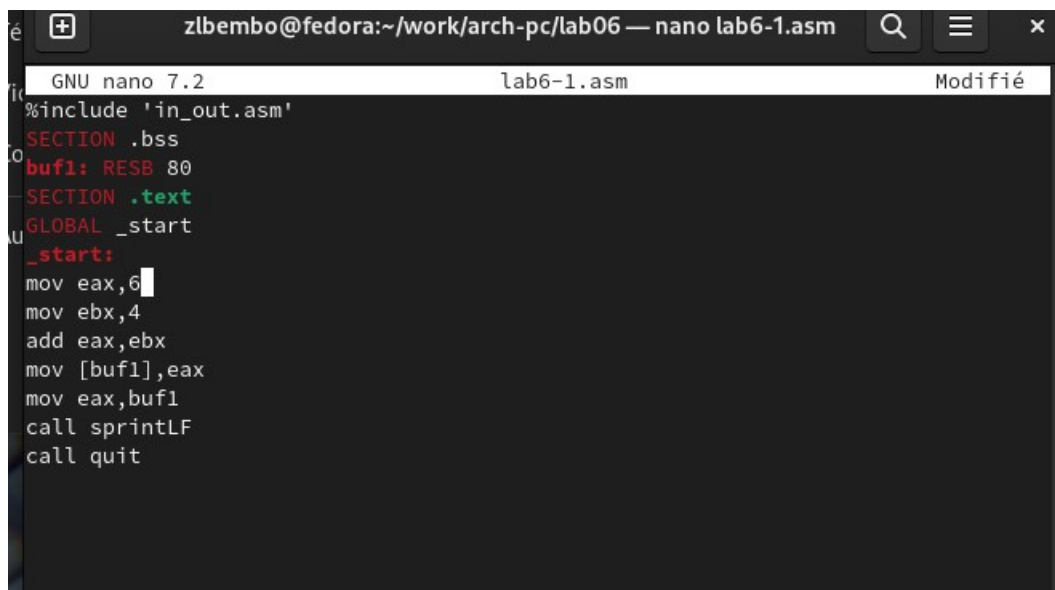
[zlbembo@fedora lab06]$ nasm -f elf lab6-1.asm
[zlbembo@fedora lab06]$ ld -m elf_i386 -o lab6-1 lab6-1.o
ld : ne peut pas trouver lab6-1.o : Aucun fichier ou dossier de ce type
[zlbembo@fedora lab06]$ ld -m elf_i386 -o lab6-1 lab6-1.o
[zlbembo@fedora lab06]$ ./lab6-1
j
[zlbembo@fedora lab06]$

```

Рис. 3.5: запуск исполняемого файла

- Изменяю в тексте программы символы “6” и “4” на цифры 6 и 4 (рис. 3.6).

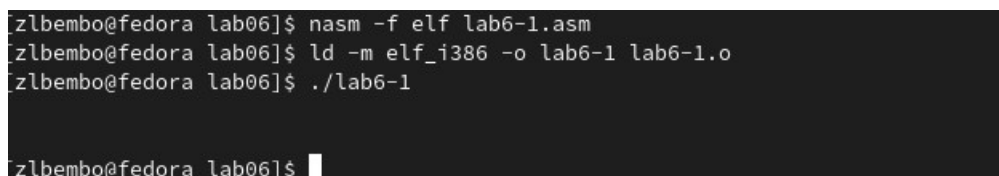




```
GNU nano 7.2 lab6-1.asm Modifié
%include 'in_out.asm'
SECTION .bss
buf1: RESB 80
SECTION .text
GLOBAL _start
_start:
mov eax,6
mov ebx,4
add eax,ebx
mov [buf1],eax
mov eax,buf1
call sprintLF
call quit
```

Рис. 3.6: редактирование файла

- Создаю новый исполняемый файл программы и запускаю его (рис. [3.7]). Теперь вывелся символ с кодом 10, это символ перевода строки, этот символ не отображается при выводе на экран.(рис. 3.7).



```
zlbembo@fedora lab06]$ nasm -f elf lab6-1.asm
zlbembo@fedora lab06]$ ld -m elf_i386 -o lab6-1 lab6-1.o
zlbembo@fedora lab06]$ ./lab6-1

zlbembo@fedora lab06]$
```

Рис. 3.7: запуск исполняемого файла

- Создаю новый файл lab6-2.asm с помощью утилиты touch (рис. 3.8).



```
[zlbembo@fedora lab06]$ touch ~/work/arch-pc/lab06/lab6-2.asm
[zlbembo@fedora lab06]$
```

Рис. 3.8: создание копии файла

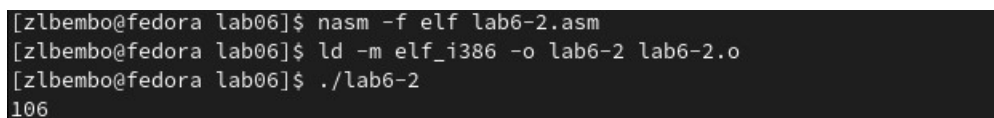
- Ввожу в файл текст другой программы для вывода значения регистра eax(рис. 3.9).



```
zlbembo@fedora:~/work/arch-pc/lab06 — nano lab6-2.asm
GNU nano 7.2 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. 3.9: редактирование файла

- Создаю и запускаю исполняемый файл lab6-2 (рис. [3.10]). Теперь вывод число 106, потому что программа позволяет вывести именно число, а не символ, хотя все еще происходит именно сложение кодов символов “6” и “4”. (рис. 3.10).



```
[zlbembo@fedora lab06]$ nasm -f elf lab6-2.asm
[zlbembo@fedora lab06]$ ld -m elf_i386 -o lab6-2 lab6-2.o
[zlbembo@fedora lab06]$ ./lab6-2
106
```

Рис. 3.10: запуск исполняемого файла

- Заменяю в тексте программы в файле lab6-2.asm символы “6” и “4” на числа 6 и 4 (рис. 3.11).



The image shows a terminal window with the nano text editor open. The title bar indicates the user is 'zlbembo' on a 'fedora' system, in the directory '~/work/arch-pc/lab06', editing the file 'lab6-2.asm'. The editor's status bar shows 'GNU nano 7.2' and the filename 'lab6-2.asm'. The code being edited is assembly language, starting with an include directive, followed by section and global declarations, and then a series of assembly instructions.

```
GNU nano 7.2 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. 3.11: редактирование файла

-Создаю и запускаю новый исполняемый файл (рис. [3.12]).. Теперь программа складывает не соответствующие символам коды в системе ASCII, а сами числа, поэтому вывод 10. (рис. 3.12).

```
[zlbembo@fedora lab06]$ nasm -f elf lab6-2.asm
[zlbembo@fedora lab06]$ ld -m elf_i386 -o lab6-2 lab6-2.o
[zlbembo@fedora lab06]$ ./lab6-2
10
[zlbembo@fedora lab06]$
```

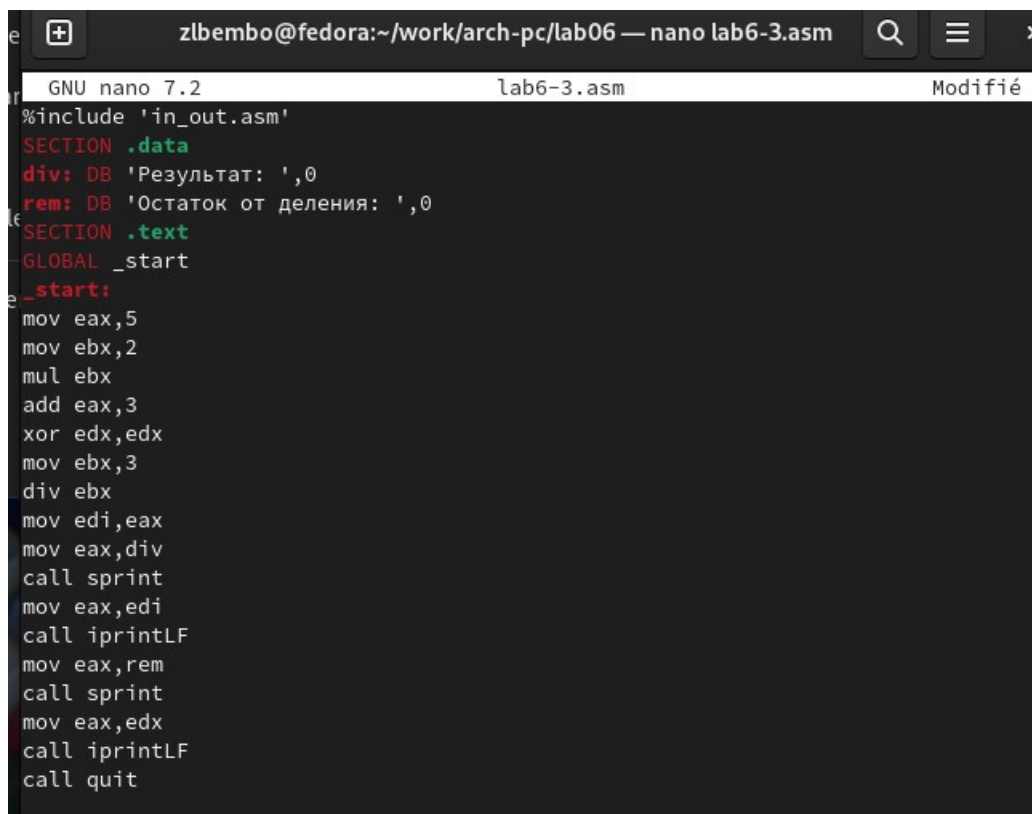
Рис. 3.12: запуск исполняемого файла

- Заменяю в тексте программы функцию `iprintLF` на `iprint`
- Создаю и запускаю новый исполняемый файл (рис. [3.13]). Вывод не изменился, потому что символ переноса строки не отображался, когда программа исполнялась с функцией `iprintLF`, а `iprint` не добавляет к выводу символ переноса строки, в отличие от `iprintLF` (рис. 3.13).

```
[zlbembo@fedora lab06]$ nasm -f elf lab6-2.asm
[zlbembo@fedora lab06]$ ld -m elf_i386 -o lab6-2 lab6-2.o
[zlbembo@fedora lab06]$ ./lab6-2
10[zlbembo@fedora lab06]$
```

Рис. 3.13: запуск исполняемого файла

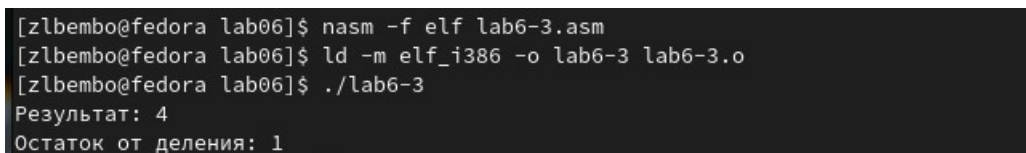
- 2) Выполнение арифметических операций в NASM
- Создаю файл `lab6-3.asm` с помощью утилиты `touch`. Ввожу в созданный файл текст программы для вычисления значения выражения  $f(x) = (5 \cdot 2 + 3)/3$  (рис. 3.14).



```
zlbembo@fedora:~/work/arch-pc/lab06 — nano lab6-3.asm
GNU nano 7.2 lab6-3.asm Modifié
%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
mov eax,edx
call iprintLF
call quit
```

Рис. 3.14: редактирование файла

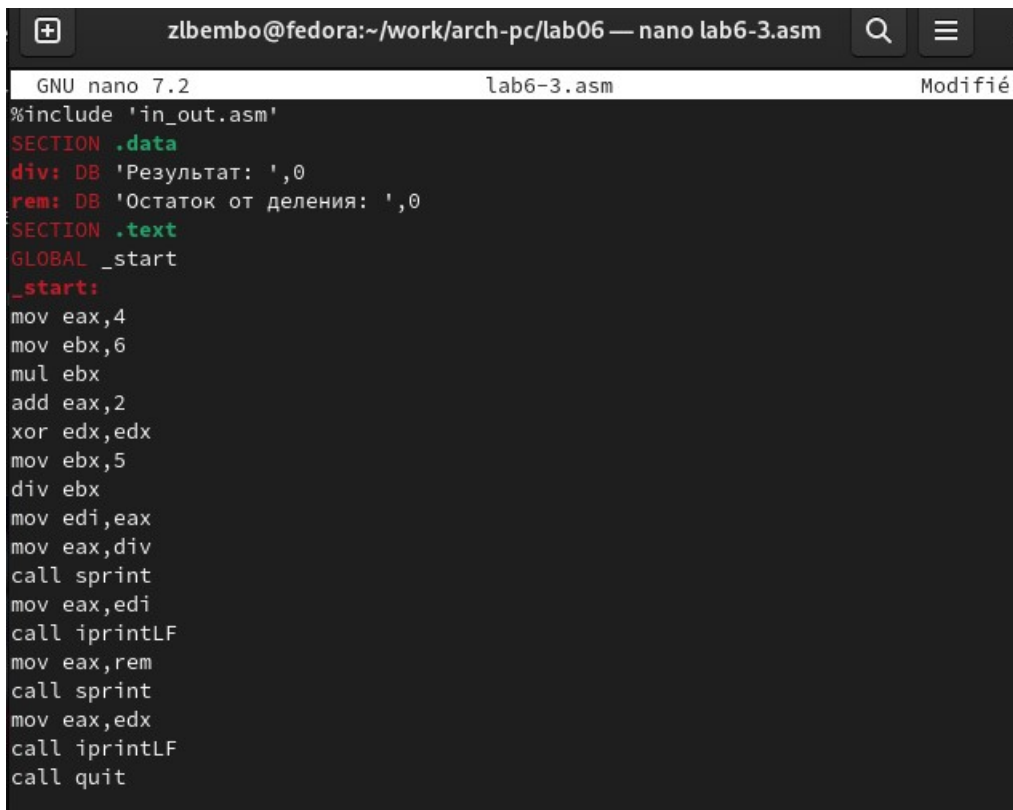
- Создаю исполняемый файл и запускаю его (рис. 3.15).



```
[zlbembo@fedora lab06]$ nasm -f elf lab6-3.asm
[zlbembo@fedora lab06]$ ld -m elf_i386 -o lab6-3 lab6-3.o
[zlbembo@fedora lab06]$ ./lab6-3
Результат: 4
Остаток от деления: 1
```

Рис. 3.15: запуск исполняемого файла

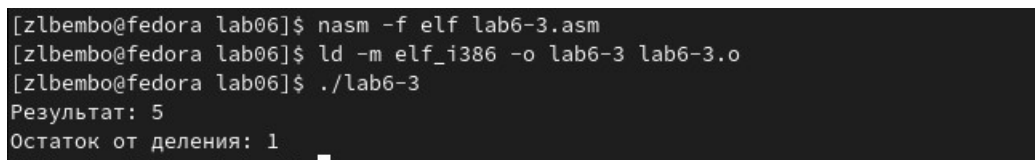
- Изменяю программу так, чтобы она вычисляла значение выражения  $f(x) = (4 \cdot 6 + 2)/5$  (рис. 3.16).



```
zlbembo@fedora:~/work/arch-pc/lab06 — nano lab6-3.asm
GNU nano 7.2 lab6-3.asm
%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
mov eax,edx
call iprintLF
call quit
```

Рис. 3.16: изменение программы

- Создаю и запускаю новый исполняемый файл (рис. [3.17]). Я посчитала для проверки правильности работы программы значение выражения самостоятельно, программа отработала верно. (рис. 3.17).



```
[zlbembo@fedora lab06]$ nasm -f elf lab6-3.asm
[zlbembo@fedora lab06]$ ld -m elf_i386 -o lab6-3 lab6-3.o
[zlbembo@fedora lab06]$ ./lab6-3
Результат: 5
Остаток от деления: 1
```

Рис. 3.17: запуск исполняемого файла

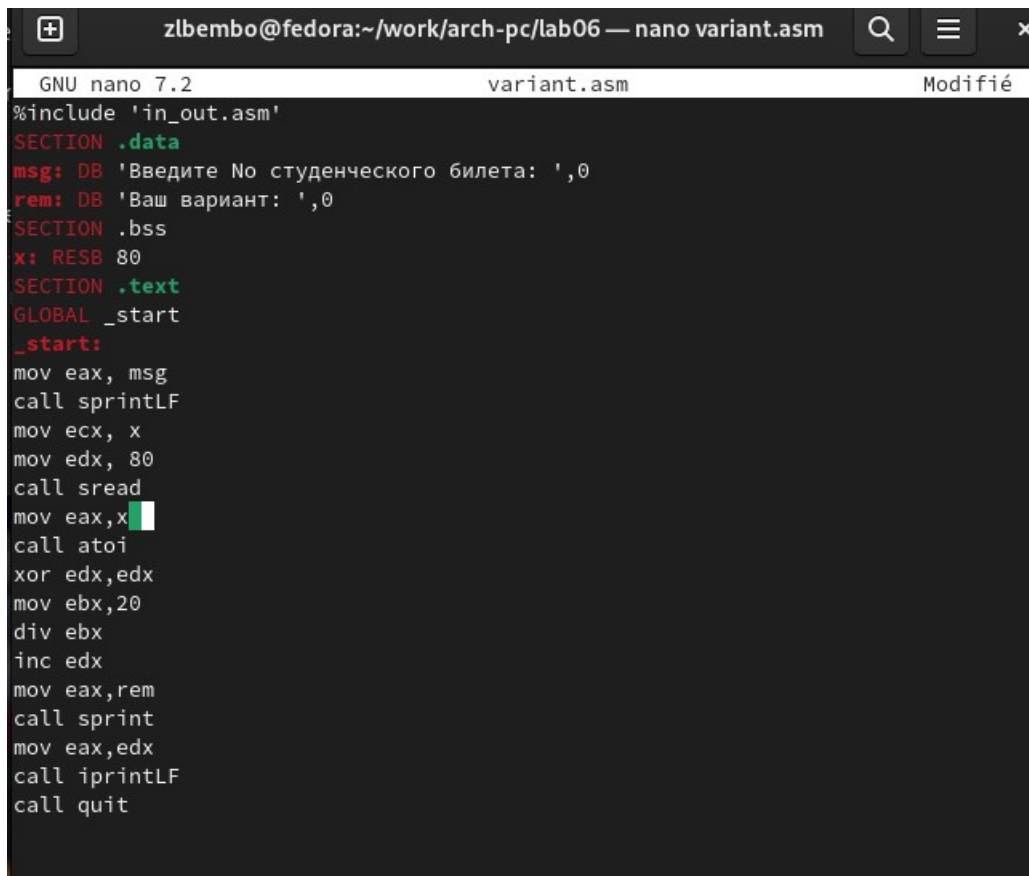
- Создаю файл variant.asm с помощью утилиты touch. (рис. 3.18).



```
[zlbembo@fedora lab06]$ touch ~/work/arch-pc/lab06/variant.asm
[zlbembo@fedora 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
```

Рис. 3.18: создание копии файла

- Ввожу в файл текст программы для вычисления варианта задания по номеру студенческого билета.(рис. 3.19).

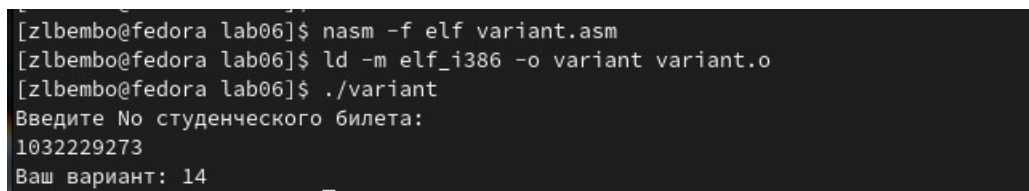


```

zlbembo@fedora:~/work/arch-pc/lab06 — nano variant.asm
GNU nano 7.2 variant.asm Modifié
#include 'in_out.asm'
SECTION .data
msg: DB 'Введите No студенческого билета: ',0
rem: DB 'Ваш вариант: ',0
SECTION .bss
x: RESB 80
SECTION .text
GLOBAL _start
_start:
mov eax, msg
call sprintLF
mov ecx, x
mov edx, 80
call sread
mov eax,x
call atoi
xor edx,edx
mov ebx,20
div ebx
inc edx
mov eax,rem
call sprint
mov eax,edx
call iprintLF
call quit
  
```

Рис. 3.19: редактирование файла

- Создаю и запускаю исполняемый файл (рис. [3.20]). Ввожу номер своего студ. билета с клавиатуры, программа вывела, что мой вариант - 15. рис. 3.20).



```

[zlbembo@fedora lab06]$ nasm -f elf variant.asm
[zlbembo@fedora lab06]$ ld -m elf_i386 -o variant variant.o
[zlbembo@fedora lab06]$ ./variant
Введите No студенческого билета:
1032229273
Ваш вариант: 14
  
```

Рис. 3.20: запуск исполняемого файла

### 3.10 Ответы на вопросы по программе

- 1) За вывод сообщения “Ваш вариант” отвечают строки кода:
  - `mov eax,rem`
  - `call sprint`
- 2) Инструкция `mov ecx, x` используется, чтобы положить адрес вводимой строки `x` в регистр `ecx` `mov edx, 80` - запись в регистр `edx` длины вводимой строки `call sread` - вызов подпрограммы из внешнего файла, обеспечивающей ввод сообщения с клавиатуры
- 3) `call atoi` используется для вызова подпрограммы из внешнего файла, которая преобразует `ascii`-код символа в целое число и записывает результат в регистр `eax`
- За вычисления варианта отвечают строки:
  - `xor edx,edx` ; обнуление `edx` для корректной работы `div`
  - `mov ebx,20` ; `ebx = 20`
  - `div ebx` ; `eax = eax/20`, `edx` - остаток от деления
  - `inc edx` ; `edx = edx + 1`
- 5) При выполнении инструкции `div ebx` остаток от деления записывается в регистр `edx`
- 6) Инструкция `inc edx` увеличивает значение регистра `edx` на 1
- 7) За вывод на экран результатов вычислений отвечают строки:
  - `mov eax,edx`
  - `call iprintLF`



- 3) Выполнение заданий для самостоятельной работы
- Создаю и запускаю исполняемый файл. рис. 3.21).

```
[zlbembo@fedora lab06]$ nasm -f elf lab6-4.asm
[zlbembo@fedora lab06]$ ld -m elf_i386 -o lab6-4 lab6-4.o
[zlbembo@fedora lab06]$ ./lab6-4
Введите значение x: 1
Результат: 140
[zlbembo@fedora lab06]$ ./lab6-4
Введите значение x: 3
Результат: 300
[zlbembo@fedora lab06]$
```

Рис. 3.21: запуск исполняемого файла

## 4 Выводы

- При выполнении данной лабораторной работы я освоила арифметические инструкции языка ассемблера NASM.

## **Список литературы**