

Лабораторная работа №6

Арифметические операции в NASM

Солдатов Алексей

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	8
4.1	Символьные и численные данные в NASM	8
4.2	Выполнение арифметических операций в NASM	13
4.3	Ответы на вопросы	19
4.4	Выполнение заданий для самостоятельной работы	20
5	Выводы	22
	Список литературы	23

Список иллюстраций

4.1	Подготовка к работе	8
4.2	Ввод команд	8
4.3	Перенос файла	9
4.4	Создание и запуск файла	10
4.5	Изменение текста программы	10
4.6	Создание файла	11
4.7	Создание файла	11
4.8	Ввод команд	11
4.9	Создание и запуск файла	12
4.10	Редактирование	12
4.11	Создание файла	12
4.12	Редактирование файла	12
4.13	Сборка файла	13
4.14	Создание нового файла	13
4.15	Ввод команд	13
4.16	Создание файла	15
4.17	Внесение изменений в код программы	15
4.18	Создание и проверка работы файла	16
4.19	Создание файла	17
4.20	Ввод текста в файл	17
4.21	Создание и проверка работы файла	18
4.22	Создание файла в каталоге	20
4.23	Написание программы	20
4.24	Запуск программы	21

Список таблиц

3.1	Описание некоторых каталогов файловой системы GNU Linux . .	7
-----	---	---

1 Цель работы

Освоить арифметические инструкции языка ассемблера NASM.

2 Задание

1. Символьные и численные данные в NASM
2. Выполнение арифметических операций в NASM
3. Ответы на вопросы
4. Выполнение заданий для самостоятельной работы

3 Теоретическое введение

Здесь описываются теоретические аспекты, связанные с выполнением работы. Например, в табл. 3.1 приведено краткое описание стандартных каталогов Unix.

Таблица 3.1: Описание некоторых каталогов файловой системы GNU Linux

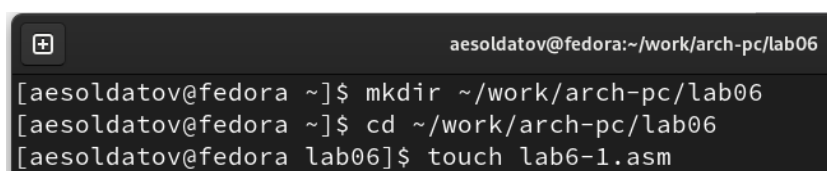
Имя каталога	Описание каталога
/	Корневая директория, содержащая всю файловую
/bin	Основные системные утилиты, необходимые как в однопользовательском режиме, так и при обычной работе всем пользователям
/etc	Общесистемные конфигурационные файлы и файлы конфигурации установленных программ
/home	Содержит домашние директории пользователей, которые, в свою очередь, содержат персональные настройки и данные пользователя
/media	Точки монтирования для сменных носителей
/root	Домашняя директория пользователя root
/tmp	Временные файлы
/usr	Вторичная иерархия для данных пользователя

Более подробно об Unix см. в [1–6].

4 Выполнение лабораторной работы

4.1 Символьные и численные данные в NASM

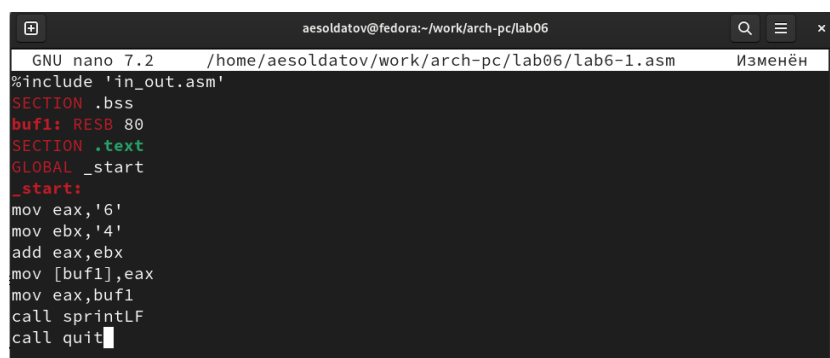
Создал каталог для программ лабораторной работы №6, перешел в него и создал файл “lab6-1.asm” (рис. [fig?];001).



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

Рис. 4.1: Подготовка к работе

Ввел в файл “lab6-1.asm” текст программы из листинга 6.1. со страницы в ТУИС (рис. 4.2).



```
GNU nano 7.2 /home/aesoldatov/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. 4.2: Ввод команд

```
%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

```

Перенес файл “in_out.asm” из прошлой папки с лабораторной работой в нынешнюю (рис. 4.3).

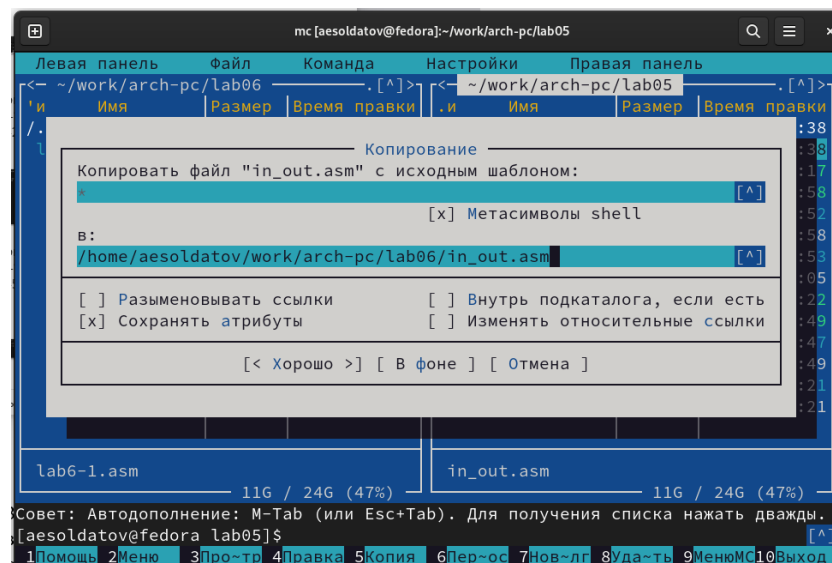


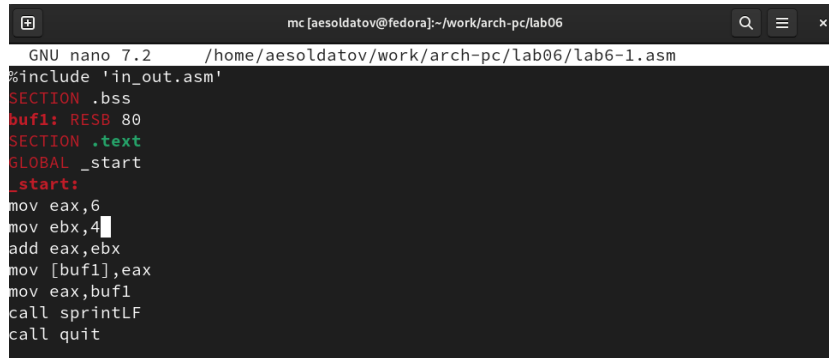
Рис. 4.3: Перенос файла

Создал исполняемый файл и запустил его (рис. 4.4).

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

Рис. 4.4: Создание и запуск файла

Далее изменил текст программы и вместо символов, записал в регистры числа (рис. 4.5).



```
mc [aesoldatov@fedora]:~/work/arch-pc/lab06
GNU nano 7.2 /home/aesoldatov/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 sprintLF
call quit
```

Рис. 4.5: Изменение текста программы

```
%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
```

Создал исполняемый файл и запустил его (рис. 4.6).

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

[aesoldatov@fedora lab06]$
```

Рис. 4.6: Создание файла

В данном случае выводится символ с кодом 10, этот символ не отображается при выводе на экран.

Создал файл “lab6-2.asm” в каталоге “~/work/arch-pc/lab06” (рис. 4.7).

```
[aesoldatov@fedora lab06]$ touch lab6-2.asm
[aesoldatov@fedora lab06]$
```

Рис. 4.7: Создание файла

Ввел в него текст программы из листинга 6.2 со страницы в ТУИС (рис. 4.8).

```
GNU nano 7.2 /home/aesoldatov/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. 4.8: Ввод команд

```
%include 'in_out.asm'
SECTION .text
GLOBAL _start
_start:
mov eax,'6'
mov ebx,'4'
add eax,ebx
call iprintLF
call quit
```

Создал исполняемый файл и запустил его (рис. 4.9).

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

Рис. 4.9: Создание и запуск файла

Исправил текст программы и вместо символов, записал в регистры числа (рис. 4.10).

```
GNU nano 7.2 /home/aesoldatov/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. 4.10: Редактирование

Создал исполняемый файл и проверил его работу (рис. 4.11).

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

Рис. 4.11: Создание файла

Заменил функцию “iprintLF” на “iprint” (рис. 4.12).

```
GNU nano 7.2 /home/aesoldatov/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. 4.12: Редактирование файла

Создал исполняемый файл и проверил его работу (рис. 4.13).

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

Рис. 4.13: Сборка файла

Подпрограмма “iprintLF” делает перевод на новую строку, а “iprint” остается на той же.

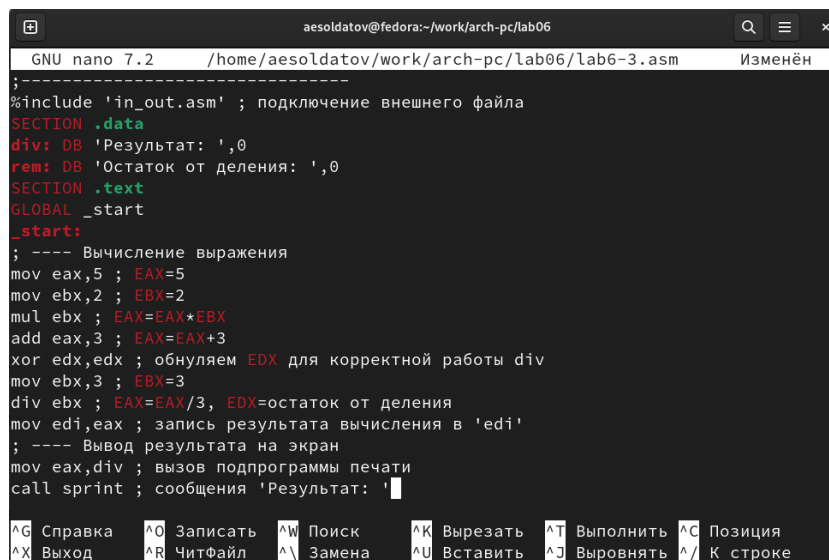
4.2 Выполнение арифметических операций в NASM

Создал файл “lab6-3.asm” в каталоге “~/work/arch-pc/lab06” (рис. 4.14).

```
[aesoldatov@fedora lab06]$ touch lab6-3.asm
[aesoldatov@fedora lab06]$
```

Рис. 4.14: Создание нового файла

Внимательно изучил текст программы из листинга 6.3 со страницы в ТУИС и ввел его в “lab6-3.asm” (рис. 4.15).



```
GNU nano 7.2 /home/aesoldatov/work/arch-pc/lab06/lab6-3.asm
;-----
%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: '
```

Рис. 4.15: Ввод команд

```

;-----
; 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

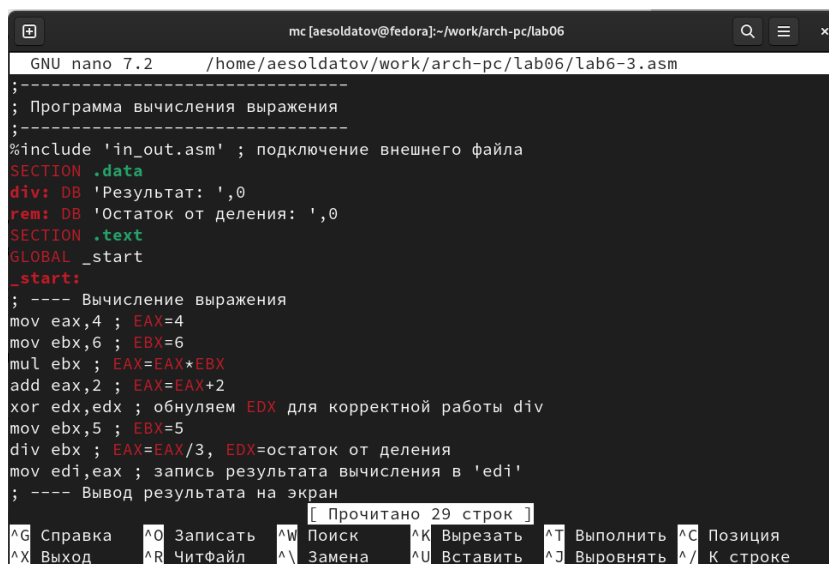
```

Создал исполняемый файл и проверил его работу, результат совпал с примером из лабораторной (рис. 4.16).

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

Рис. 4.16: Создание файла

Изменил текст программы для вычисления выражения $f(x) = (4 * 6 + 2)/5$ (рис. 4.17).



```
GNU nano 7.2 /home/aesoldatov/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/3, EDX=остаток от деления
mov edi,eax ; запись результата вычисления в 'edi'
; ---- Вывод результата на экран
[ Прочитано 29 строк ]
^G Справка ^O Записать ^W Поиск ^K Вырезать ^T Выполнить ^C Позиция
^X Выход ^R ЧитФайл ^\ Замена ^U Вставить ^J Вывернуть ^_ К строке
```

Рис. 4.17: Внесение изменений в код программы

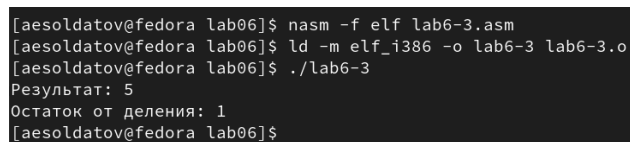
```
;-----
; Программа вычисления выражения
;-----
%include 'in_out.asm' ; подключение внешнего файла
SECTION .data
div: DB 'Результат: ',0
rem: DB 'Остаток от деления: ',0
```

```

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/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

```

Создал исполняемый файл и проверил его работу (рис. 4.18).



```

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

```

Рис. 4.18: Создание и проверка работы файла

Создал файл “variant.asm” в каталоге “~/work/arch-pc/lab06” (рис. 4.19).


```
[aesoldatov@fedora lab06]$ touch variant.asm
[aesoldatov@fedora lab06]$
```

Рис. 4.19: Создание файла

Внимательно изучил текст программы из листинга 6.4 на странице в ТУИС и ввел его в файл variant.asm (рис. 4.20).

```
GNU nano 7.2 /home/aesoldatov/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. 4.20: Ввод текста в файл

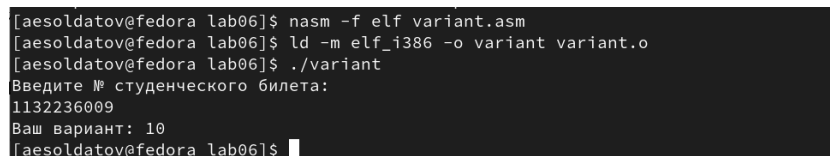
```
;-----
; Программа вычисления варианта
;-----
#include 'in_out.asm'
SECTION .data
msg: DB 'Введите № студенческого билета: ',0
rem: DB 'Ваш вариант: ',0
SECTION .bss
```

```

x: RESB 80
SECTION .text
GLOBAL _start
_start:
mov eax, msg
call sprintLF
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
mov eax,edx
call iprintLF
call quit

```

Создал исполняемый файл и проверил его работу, программа выдала, что мой номер 10 (Ответ совпадает с аналитическим решением) (рис. 4.21).



```

[aesoldatov@fedora lab06]$ nasm -f elf variant.asm
[aesoldatov@fedora lab06]$ ld -m elf_i386 -o variant variant.o
[aesoldatov@fedora lab06]$ ./variant
Введите № студенческого билета:
1132236009
Ваш вариант: 10
[aesoldatov@fedora lab06]$

```

Рис. 4.21: Создание и проверка работы файла

4.3 Ответы на вопросы

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”.
4. За вычисления варианта отвечают строки:

```
хот edx, edx ; обнуление eax 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
```

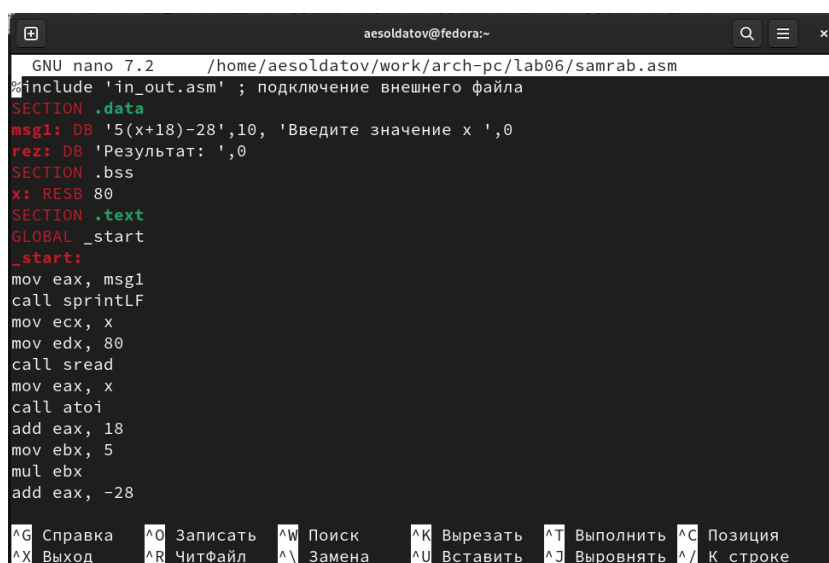
4.4 Выполнение заданий для самостоятельной работы

Создаю файл “samrab.asm” (рис. 4.22).

```
[aesoldatov@fedora lab06]$ touch samrab.asm
```

Рис. 4.22: Создание файла в каталоге

Пишу в нем программу по варианту, вычисленному из предыдущей программы (рис. 4.23).



```
GNU nano 7.2 /home/aesoldatov/work/arch-pc/lab06/samrab.asm
#include 'in_out.asm' ; подключение внешнего файла
SECTION .data
msg1: DB '5(x+18)-28',10, 'Введите значение x ',0
rez: DB 'Результат: ',0
SECTION .bss
x: RESB 80
SECTION .text
GLOBAL _start
_start:
mov eax, msg1
call sprintf
mov ecx, x
mov edx, 80
call sread
mov eax, x
call atoi
add eax, 18
mov ebx, 5
mul ebx
add eax, -28
```

Рис. 4.23: Написание программы

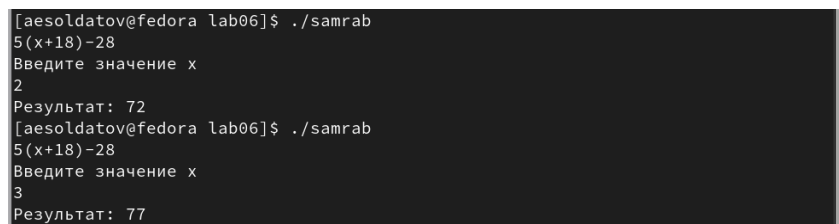
```
%include 'in_out.asm' ; подключение внешнего файла
SECTION .data
msg1: DB '5(x+18)-28',10, 'Введите значение x ',0
rez: DB 'Результат: ',0
SECTION .bss
x: RESB 80
SECTION .text
GLOBAL _start
```

```

_start:
mov eax, msg1
call sprintf
mov ecx, x
mov edx, 80
call sread
mov eax, x
call atoi
add eax, 18
mov ebx, 5
mul ebx
add eax, -28
mov edi, eax
mov eax, rez
call sprintf
mov eax, edi
call iprintLF
call quit

```

Создаю исполняемый файл и запускаю программу (рис. 4.24).



```

[aesoldatov@fedora lab06]$ ./samrab
5(x+18)-28
Введите значение x
2
Результат: 72
[aesoldatov@fedora lab06]$ ./samrab
5(x+18)-28
Введите значение x
3
Результат: 77

```

Рис. 4.24: Запуск программы

Программа выдает правильный ответ.

5 Выводы

Освоил арифметические инструкции языка ассемблера NASM.

Список литературы

1. GNU Bash Manual [Электронный ресурс]. Free Software Foundation, 2016.
URL: <https://www.gnu.org/software/bash/manual/>.
2. Newham C. Learning the bash Shell: Unix Shell Programming. O'Reilly Media, 2005. 354 с.
3. Zarrelli G. Mastering Bash. Packt Publishing, 2017. 502 с.
4. Robbins A. Bash Pocket Reference. O'Reilly Media, 2016. 156 с.
5. Таненбаум Э. Архитектура компьютера. 6-е изд. СПб.: Питер, 2013. 874 с.
6. Таненбаум Э., Бос Х. Современные операционные системы. 4-е изд. СПб.: Питер, 2015. 1120 с.