

Лабораторная работа №7

Архитектура компьютера

Голованова Мария Константиновна

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
3.1	Символьные и численные данные в NASM	7
3.2	Выполнение арифметических операций в NASM	12
4	Выполнение самостоятельной работы	17
5	Выводы	21

Список иллюстраций

3.1	Создание каталога для программ лабораторной работы №7 и файла lab7-1.asm	7
3.2	Введение текста программы из листинга 7.1	8
3.3	Создание и запуск исполняемого файла lab7-1	8
3.4	Исправление текста файла lab7-1.asm	9
3.5	Создание и запуск исполняемого исправленного файла lab7-1	9
3.6	Создание файла lab7-2.asm	9
3.7	Введение текста программы из листинга 7.2	10
3.8	Создание и запуск исполняемого файла lab7-2	10
3.9	Исправление текста файла lab7-2.asm	11
3.10	Создание и запуск исполняемого исправленного файла lab7-2	11
3.11	Замена функции iprintLF на iprint в тексте файла lab7-2.asm	12
3.12	Создание и запуск исполняемого исправленного файла lab7-2	12
3.13	Создание файла lab7-3.asm	12
3.14	Введение текста программы из листинга 7.3	13
3.15	Создание и запуск исполняемого файла lab7-3	13
3.16	Исправление текста файла lab7-3.asm	14
3.17	Создание и запуск исполняемого изменённого файла lab7-3	14
3.18	Создание файла variant.asm	15
3.19	Введение текста программы из листинга 7.4	15
3.20	Создание и запуск исполняемого файла variant	16
4.1	Создание файла var19.asm	17
4.2	Введение текста программы для функции из таблицы 7.3	18
4.3	Таблица 7.3	19
4.4	Создание и проверка исполняемого файла var19	20

Список таблиц

1 Цель работы

Освоение арифметических инструкций языка ассемблера NASM.

2 Задание

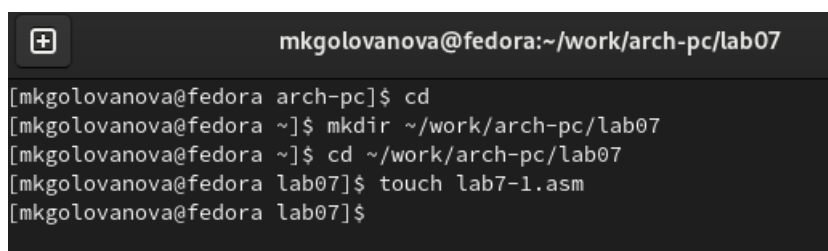
Создать программы вывода символьных и численных значений, программы вычисления арифметических выражений. Создать программу вычисления выражения $y = f(x)$, выбрав вариант из таблицы 7.3 по номеру, полученному в при выполнении лабораторной работы.

3 Теоретическое введение

Арифметические инструкции языка ассемблера NASM - операции, позволяющие производить арифметические вычисления в ассемблере. К ним относятся такие операции как команда целочисленного сложения `add`, команда целочисленного вычитания `sub`, команды инкремента и декремента `inc` (от англ. *increment*) и `dec` (от англ. *decrement*), команда изменения знака операнда `neg`, команды умножения `mul` и `imul`, команды деления `div` и `idiv`. # Выполнение лабораторной работы

3.1 Символьные и численные данные в NASM

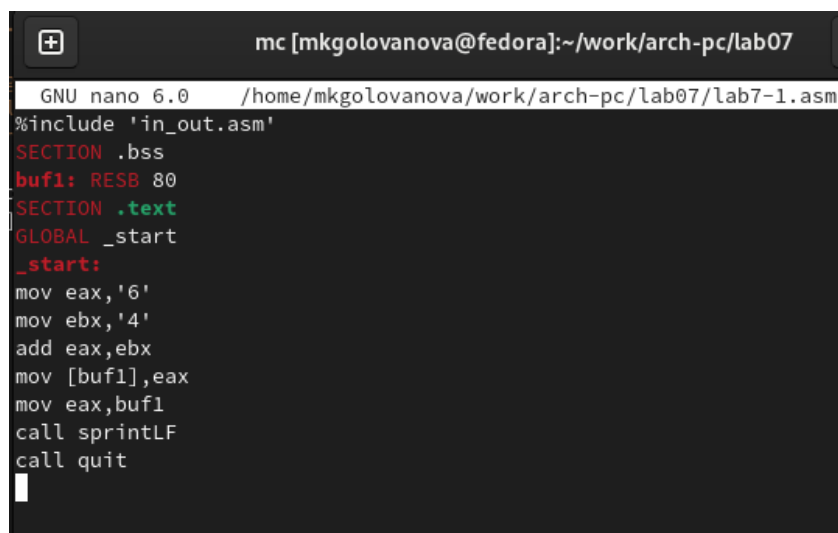
Я создала каталог для программ лабораторной работы №7, перешла в него и создала файл `lab7-1.asm` (рис. 3.1).



```
mkgolovanova@fedora:~/work/arch-pc/lab07
[mkgolovanova@fedora arch-pc]$ cd
[mkgolovanova@fedora ~]$ mkdir ~/work/arch-pc/lab07
[mkgolovanova@fedora ~]$ cd ~/work/arch-pc/lab07
[mkgolovanova@fedora lab07]$ touch lab7-1.asm
[mkgolovanova@fedora lab07]$
```

Рис. 3.1: Создание каталога для программ лабораторной работы №7 и файла `lab7-1.asm`

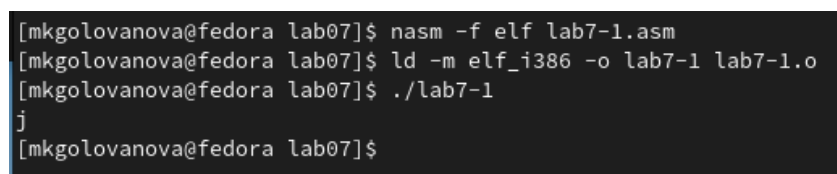
Я ввела в файл `lab7-1.asm` текст программы из листинга 7.1 (рис. 3.2).



```
mc [mkgolovanova@fedora]:~/work/arch-pc/lab07
GNU nano 6.0 /home/mkgolovanova/work/arch-pc/lab07/lab7-1.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. 3.2: Введение текста программы из листинга 7.1

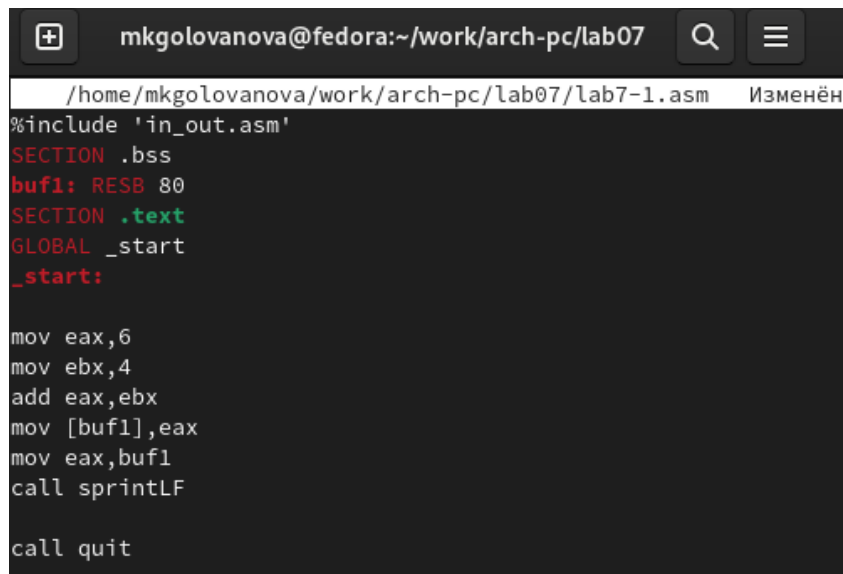
Я создала исполняемый файл и запустила его (рис. 3.3).



```
[mkgolovanova@fedora lab07]$ nasm -f elf lab7-1.asm
[mkgolovanova@fedora lab07]$ ld -m elf_i386 -o lab7-1 lab7-1.o
[mkgolovanova@fedora lab07]$ ./lab7-1
j
[mkgolovanova@fedora lab07]$
```

Рис. 3.3: Создание и запуск исполняемого файла lab7-1

В данном случае при выводе значения регистра `eax` я ожидала увидеть число 10. Однако результатом будет символ `j`. Это происходит потому, что код символа `б` равен 00110110 в двоичном представлении (или 54 в десятичном представлении), а код символа `4` – 00110100 (52). Команда `add eax,ebx` записала в регистр `eax` сумму кодов – 01101010 (106), что в свою очередь является кодом символа `j`. Я исправила текст программы и вместо символов записала в регистры числа, заменив строки `mov eax,'6'` `mov ebx,'4'` на строки `mov eax,6` `mov ebx,4` (рис. 3.4).



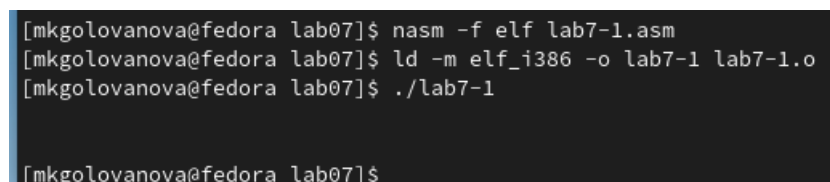
```
mkgolovanova@fedora:~/work/arch-pc/lab07
/home/mkgolovanova/work/arch-pc/lab07/lab7-1.asm  Изменён
%include 'in_out.asm'
SECTION .bss
buf1: RESB 80
SECTION .text
GLOBAL _start
_start:

mov eax,6
mov ebx,4
add eax,ebx
mov [buf1],eax
mov eax,buf1
call sprintLF

call quit
```

Рис. 3.4: Исправление текста файла lab7-1.asm

Я создала исполняемый файл и запустила его (рис. 3.5).



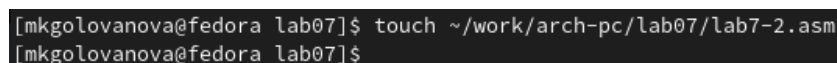
```
[mkgolovanova@fedora lab07]$ nasm -f elf lab7-1.asm
[mkgolovanova@fedora lab07]$ ld -m elf_i386 -o lab7-1 lab7-1.o
[mkgolovanova@fedora lab07]$ ./lab7-1

[mkgolovanova@fedora lab07]$
```

Рис. 3.5: Создание и запуск исполняемого исправленного файла lab7-1

В данном случае выводится символ с кодом 10. Пользуясь таблицей ASCII, я определила, что код 10 соответствует символу 'LF, /n', причём символ не отображается при выводе на экран.

Я создала файл lab7-2.asm в каталоге ~/work/arch-pc/lab07 и ввела в него текст программы из листинга 7.2 (рис. 3.6, рис. 3.7).



```
[mkgolovanova@fedora lab07]$ touch ~/work/arch-pc/lab07/lab7-2.asm
[mkgolovanova@fedora lab07]$
```

Рис. 3.6: Создание файла lab7-2.asm



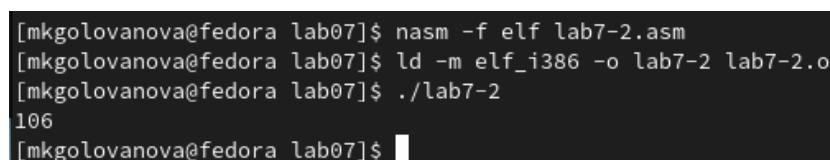
```
GNU nano 6.0 /home/mkgolovanova/work/arch-pc/lab07/lab7-2.asm Изм
%include 'in_out.asm'
SECTION .text
GLOBAL _start
_start:

mov eax, '6'
mov ebx, '4'
add eax, ebx
call iprintLF

call quit
```

Рис. 3.7: Введение текста программы из листинга 7.2

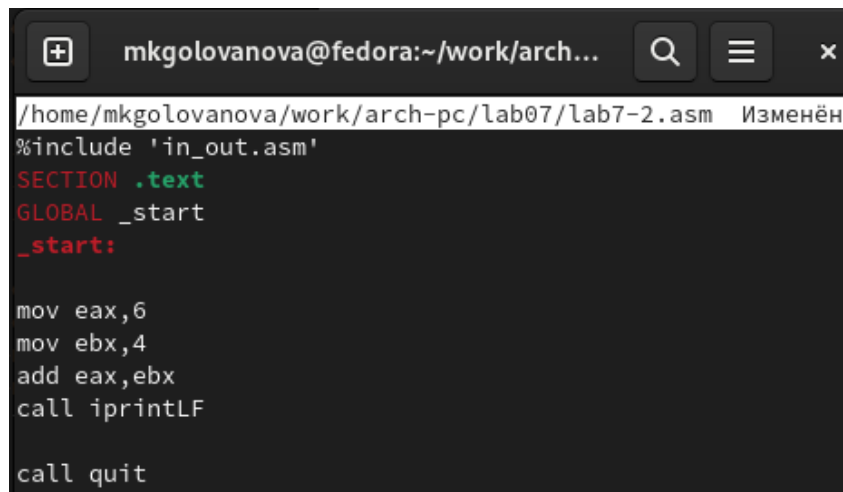
Я создала исполняемый файл и запустила его (рис. 3.8).



```
[mkgolovanova@fedora lab07]$ nasm -f elf lab7-2.asm
[mkgolovanova@fedora lab07]$ ld -m elf_i386 -o lab7-2 lab7-2.o
[mkgolovanova@fedora lab07]$ ./lab7-2
106
[mkgolovanova@fedora lab07]$
```

Рис. 3.8: Создание и запуск исполняемого файла lab7-2

В результате работы программы я получила 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. Я изменила символы на числа, заменив строки `mov eax, '6'` `mov ebx, '4'` на строки `mov eax, 6` `mov ebx, 4` (рис. 3.9).



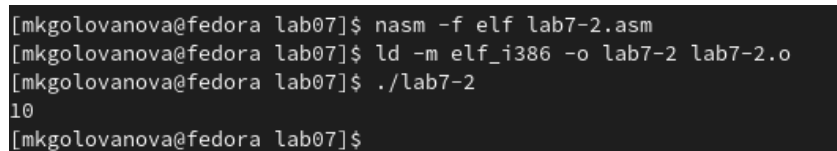
```
mkgolovanova@fedora:~/work/arch...
/home/mkgolovanova/work/arch-pc/lab07/lab7-2.asm  Изменён
#include 'in_out.asm'
SECTION .text
GLOBAL _start
_start:

mov eax,6
mov ebx,4
add eax,ebx
call iprintLF

call quit
```

Рис. 3.9: Исправление текста файла lab7-2.asm

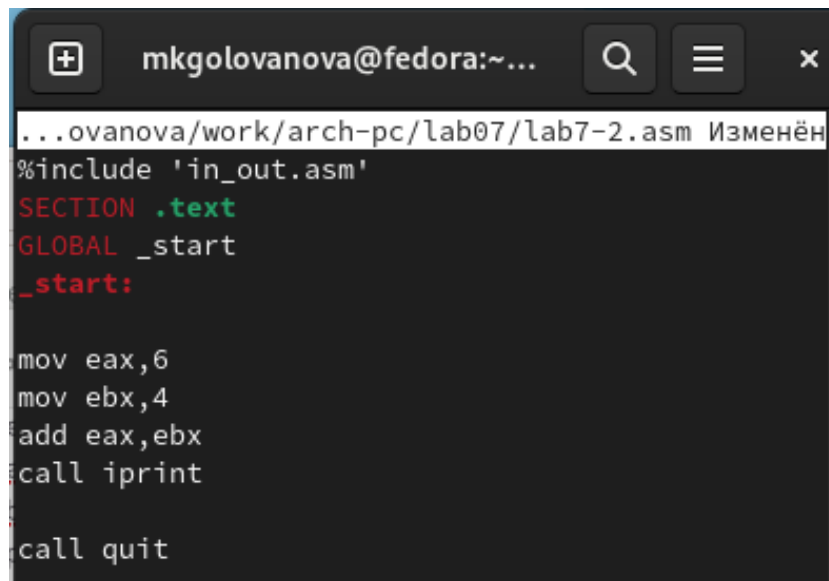
Я создала исполняемый файл и запустила его (рис. 3.10).



```
[mkgolovanova@fedora lab07]$ nasm -f elf lab7-2.asm
[mkgolovanova@fedora lab07]$ ld -m elf_i386 -o lab7-2 lab7-2.o
[mkgolovanova@fedora lab07]$ ./lab7-2
10
[mkgolovanova@fedora lab07]$
```

Рис. 3.10: Создание и запуск исполняемого исправленного файла lab7-2

В результате работы программы я получила число 10. Я заменила функцию iprintLF на iprint, создала исполняемый файл и запустила его (рис. 3.11, рис. 3.12). После вывода функция iprint в отличие от функции iprintLF не переводит строку.

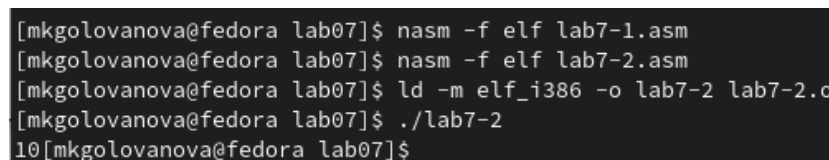


```
..ovanova/work/arch-pc/lab07/lab7-2.asm Изменён
%include 'in_out.asm'
SECTION .text
GLOBAL _start
_start:

mov eax,6
mov ebx,4
add eax,ebx
call iprint

call quit
```

Рис. 3.11: Замена функции `iprintLF` на `iprint` в тексте файла `lab7-2.asm`



```
[mkgolovanova@fedora lab07]$ nasm -f elf lab7-1.asm
[mkgolovanova@fedora lab07]$ nasm -f elf lab7-2.asm
[mkgolovanova@fedora lab07]$ ld -m elf_i386 -o lab7-2 lab7-2.o
[mkgolovanova@fedora lab07]$ ./lab7-2
10[mkgolovanova@fedora lab07]$
```

Рис. 3.12: Создание и запуск исполняемого исправленного файла `lab7-2`

3.2 Выполнение арифметических операций в NASM

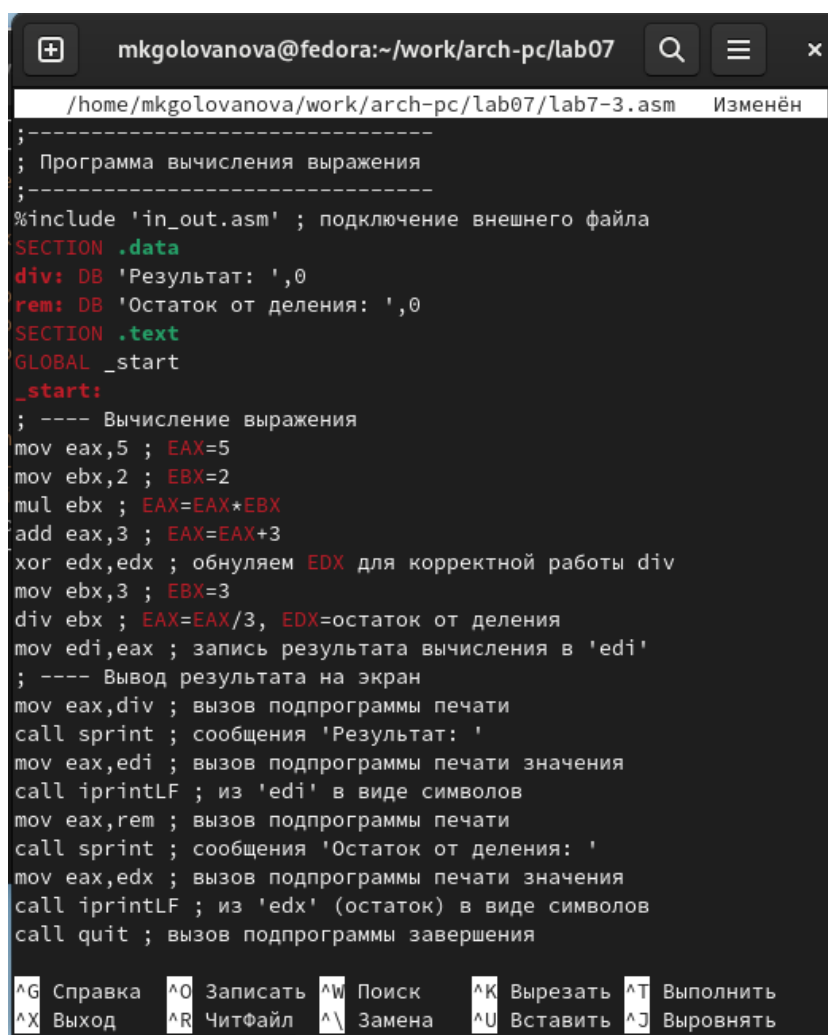
В качестве примера выполнения арифметических операций в NASM я рассмотрела программу вычисления арифметического выражения $f(x) = (5 * 2 + 3)/3$. Я создала файл `lab7-3.asm` в каталоге `~/work/arch-pc/lab07` (рис. 3.13).



```
[mkgolovanova@fedora lab07]$ touch ~/work/arch-pc/lab07/lab7-3.asm
[mkgolovanova@fedora lab07]$
```

Рис. 3.13: Создание файла `lab7-3.asm`

Я внимательно изучила текст программы из листинга 7.3 и ввела в `lab7-3.asm` (рис. 3.14)

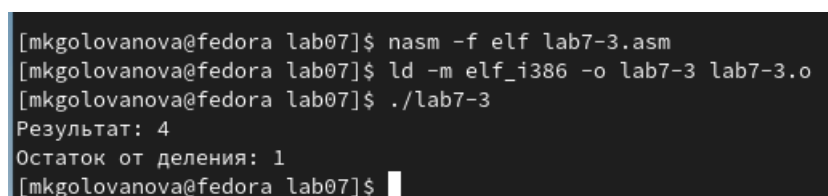


```
mkgolovanova@fedora:~/work/arch-pc/lab07
/home/mkgolovanova/work/arch-pc/lab07/lab7-3.asm Изменён
;-----
; Программа вычисления выражения
;-----
%include 'in_out.asm' ; подключение внешнего файла
SECTION .data
div: DB 'Результат: ',0
rem: DB 'Остаток от деления: ',0
SECTION .text
GLOBAL _start
_start:
; ---- 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

^G Справка ^O Записать ^W Поиск ^K Вырезать ^T Выполнить
^X Выход ^R ЧитФайл ^\ Замена ^U Вставить ^J Выровнять
```

Рис. 3.14: Введение текста программы из листинга 7.3

Я создала исполняемый файл и запустила его (рис. 3.15).



```
[mkgolovanova@fedora lab07]$ nasm -f elf lab7-3.asm
[mkgolovanova@fedora lab07]$ ld -m elf_i386 -o lab7-3 lab7-3.o
[mkgolovanova@fedora lab07]$ ./lab7-3
Результат: 4
Остаток от деления: 1
[mkgolovanova@fedora lab07]$
```

Рис. 3.15: Создание и запуск исполняемого файла lab7-3

Я изменила текст программы для вычисления выражения $f(x) = (4 * 6 + 2)/5$, создала исполняемый файл и проверила его работу (рис. 3.16, рис. 3.17).

```

mkgolovanova@fedora:~/work/arch-pc/lab07
GNU nano 6.0 /home/mkgolovanova/work/arch-pc/lab07/lab7-3.
;-----
; 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. 3.16: Исправление текста файла lab7-3.asm

```

[mkgolovanova@fedora lab07]$ nasm -f elf lab7-3.asm
[mkgolovanova@fedora lab07]$ ld -m elf_i386 -o lab7-3 lab7-3.o
[mkgolovanova@fedora lab07]$ ./lab7-3
Результат: 5
Остаток от деления: 1
[mkgolovanova@fedora lab07]$

```

Рис. 3.17: Создание и запуск исполняемого изменённого файла lab7-3

В качестве другого примера я рассмотрела программу вычисления варианта задания по номеру студенческого билета, работающую по следующему алгоритму:

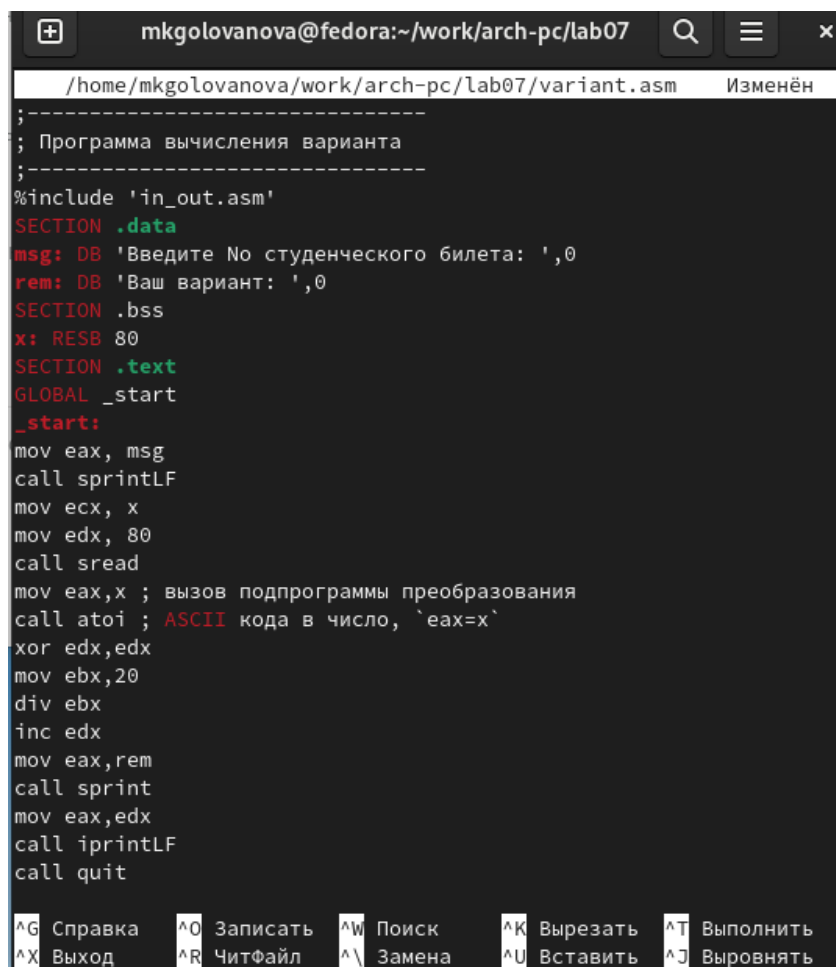
- вывести запрос на введение No студенческого билета;
- вычислить номер варианта по формуле: $(S_n \bmod 20) + 1$, где S_n – номер студенческого билета (В данном случае $a \bmod b$ – это остаток от деления a на b);
- вывести на экран номер

варианта. Я создала файл variant.asm в каталоге ~/work/arch-pc/lab07 (рис. 3.18).

```
[mkgolovanova@fedora lab07]$ touch ~/work/arch-pc/lab07/variant.asm
[mkgolovanova@fedora lab07]$
```

Рис. 3.18: Создание файла variant.asm

Внимательно изучите текст программы из листинга 7.4 и введите в файл variant.asm (рис. 3.19).



```
mkgolovanova@fedora:~/work/arch-pc/lab07
/home/mkgolovanova/work/arch-pc/lab07/variant.asm  Изменён
;-----
; Программа вычисления варианта
;-----
%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 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

^G Справка  ^O Записать  ^W Поиск    ^K Вырезать  ^T Выполнить
^X Выход    ^R ЧитФайл  ^\ Замена  ^U Вставить  ^J Выворнять
```

Рис. 3.19: Введение текста программы из листинга 7.4

Я создала исполняемый файл и запустила его (рис. 3.20).

```

[mkgolovanova@fedora lab07]$ nasm -f elf variant.asm
[mkgolovanova@fedora lab07]$ ld -m elf_i386 -o variant variant.o
[mkgolovanova@fedora lab07]$ ./variant
Введите No студенческого билета:
1132226478
Ваш вариант: 19
[mkgolovanova@fedora lab07]$

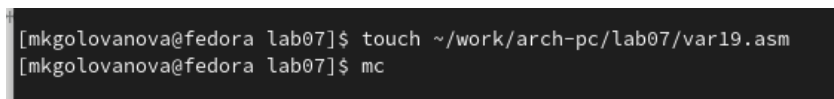
```

Рис. 3.20: Создание и запуск исполняемого файла variant

Я проверила результат работы программы, вычислив номер варианта аналитически. Ответы на вопросы: 1. За вывод на экран сообщения ‘Ваш вариант:’ отвечают строки: `mov eax,rem call sprint` 2. Инструкции `nasm: mov ecx, x` означает запись адреса переменной в ‘EAX’, `mov edx, 80` - запись длины вводимого сообщения в ‘EBX’, `call sread` - вызов подпрограммы ввода сообщения. 3. Инструкция “`call atoi`” используется для преобразования ASCII кода в число. 4. За вычисления варианта отвечают строки: `xor edx,edx mov ebx,20 div ebx inc edx` 5. Остаток от деления при выполнении инструкции “`div ebx`” записывается в регистр `edx`. 6. Инструкция “`inc edx`” используется для прибавления к числу единицы. 7. За вывод на экран результата вычислений отвечают строки: `mov eax,edx call iprintLF`

4 Выполнение самостоятельной работы

Я создала файл var19.asm и написала в нём программу вычисления выражения $y = f(x)$ так, чтобы она выводила выражение для вычисления, выводила запрос на ввод значения x , вычисляла заданное выражение в зависимости от введенного x , выводила результат вычислений (рис. 4.1, рис. 4.2). Вид функции $f(x)$ я выбрала из таблицы 7.3 вариантов заданий в соответствии с номером, полученным мной при выполнении лабораторной работы (Вариант №19: $(1/3 * x + 5) * 7$, $x_1=3$, $x_2=9$)(рис. 4.3).

A screenshot of a terminal window with a dark background and light-colored text. It shows two lines of commands being executed in a shell. The first line uses the 'touch' command to create a file named 'var19.asm' in the directory '~/.work/arch-pc/lab07/'. The second line shows the prompt for the next command.

```
[mkgolovanova@fedora lab07]$ touch ~/.work/arch-pc/lab07/var19.asm
[mkgolovanova@fedora lab07]$ mc
```

Рис. 4.1: Создание файла var19.asm

```
mkgolovanova@fedora:~/work/arch-pc/lab07
GNU nano 6.0 /home/mkgolovanova/work/arch-pc/lab07/var19.asm
;-----
; Программа вычисления выражения  $y = f(x)$ , вариант 19
;-----
#include 'in_out.asm' ; подключение внешнего файла
SECTION .data
msg: DB 'Введите число: ',0
rem: DB 'Результат: ',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 ; обнуляем EDX для корректной работы div
mov ebx,3 ; EBX=3
div ebx ; EAX=EAX/3, EDX=остаток от деления
add eax,5 ; EAX=EAX+5
mov ebx,7 ; EBX=7
mul ebx ; EAX=EAX*EBX
mov edi,eax ; запись результата вычисления в 'edi'

mov eax,rem ; вызов подпрограммы печати
call sprint ; сообщения 'Результат: '
mov eax,edi ; вызов подпрограммы печати значения
call iprintLF ; из 'edi' в виде символов

call quit ; вызов подпрограммы завершения
```

Рис. 4.2: Введение текста программы для функции из таблицы 7.3

Номер варианта	Выражение для $f(x)$	x_1	x_2
1	$(10 + 2x)/3$	1	10
2	$(12x + 3)5$	1	6
3	$(2 + x)^2$	2	8
4	$\frac{4}{3}(x - 1) + 5$	4	10
5	$(9x - 8)/8$	8	64
6	$x^3/2 + 1$	2	5
7	$5(x - 1)^2$	3	5
8	$(11 + x) \cdot 2 - 6$	1	9
9	$10 + (31x - 5)$	3	1
10	$5(x + 18) - 28$	2	3
11	$10(x + 1) - 10$	1	7
12	$(8x - 6)/2$	1	5
13	$(8x + 6) \cdot 10$	1	4
14	$(\frac{x}{2} + 8) \cdot 3$	1	4
15	$(5 + x)^2 - 3$	5	1
16	$(10x - 5)^2$	3	1
17	$18(x + 1)/6$	3	1
18	$3(x + 10) - 20$	1	5
19	$(\frac{1}{3}x + 5) \cdot 7$	3	9
20	$x^3 \cdot \frac{1}{3} + 21$	1	3

Рис. 4.3: Таблица 7.3

Я создала исполняемый файл и проверила его работу для значений x_1 и x_2 из таблицы 7.3 (рис. 4.4).

```
[mkgolovanova@fedora lab07]$ nasm -f elf var19.asm
[mkgolovanova@fedora lab07]$ ld -m elf_i386 -o var19 var19.o
[mkgolovanova@fedora lab07]$ ./var19
Введите число:
3
Результат: 42
[mkgolovanova@fedora lab07]$ ./var19
Введите число:
9
Результат: 56
[mkgolovanova@fedora lab07]$
```

Рис. 4.4: Создание и проверка исполняемого файла var19

5 Выводы

Я освоила арифметические инструкции языка ассемблера NASM.