

Отчёт по лабораторной работе №6

Дисциплина: архитектура компьютера

Мусатова Екатерина Викторовна

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Ответы на вопросы по программе	14
4	Самостоятельная работа	15
5	Листинг 6.5. Программа для вычисления значения выражения $(11 + x) * 2 - 6$.	18
6	Выводы	19

Список иллюстраций

2.1	Создание необходимого каталога и файла	6
2.2	Ввод программы	7
2.3	Проверка	7
2.4	Изменение текста программы	8
2.5	Проверка программы	8
2.6	Создание файла	9
2.7	Программа вывода значения регистра еах	9
2.8	Проверка программы	9
2.9	Изменение программы	10
2.10	Проверка	10
2.11	Замена функции	11
2.12	Проверка программы	11
2.13	Создание нового файла	11
2.14	Проверка программы	12
2.15	Изменение программы	12
2.16	Проверка	12
2.17	Создание нового файла	13
2.18	Создание программы	13
2.19	Проверка работы программы	13
4.1	Создание файла	15
4.2	Создание программы	16
4.3	Создание исполняемого файла	16
4.4	Проверка работы программы	16
4.5	Проверка 2	17

Список таблиц

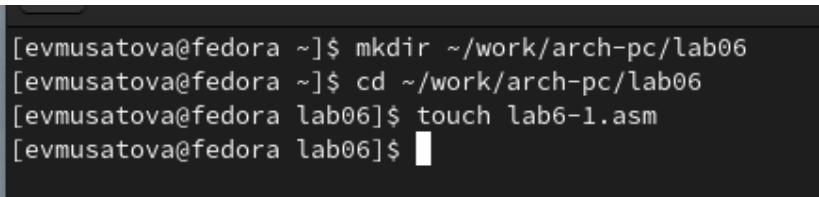
1 Цель работы

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

2 Выполнение лабораторной работы

1

Создайте каталог для программ лабораторной работы № 6, перейдя в него и создав файл lab6-1.asm (рис. 2.1).

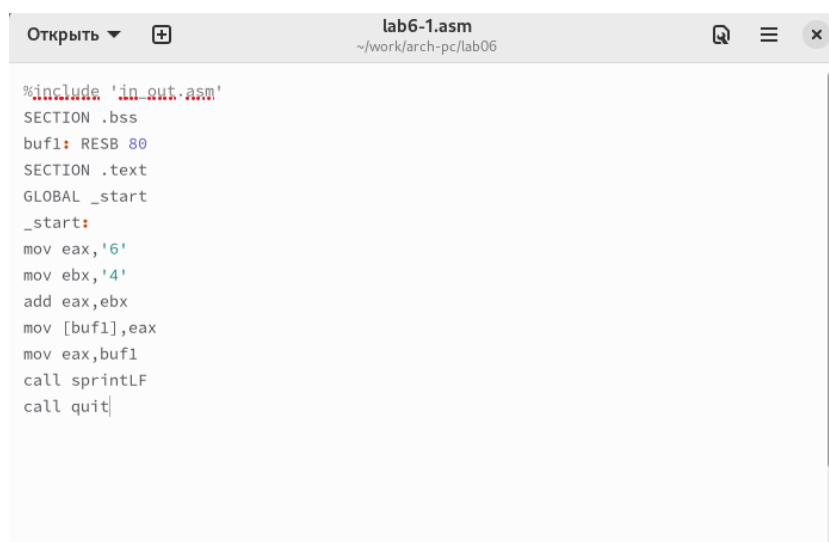


```
[evmusatova@fedora ~]$ mkdir ~/work/arch-pc/lab06
[evmusatova@fedora ~]$ cd ~/work/arch-pc/lab06
[evmusatova@fedora lab06]$ touch lab6-1.asm
[evmusatova@fedora lab06]$
```

Рис. 2.1: Создание необходимого каталога и файла

2

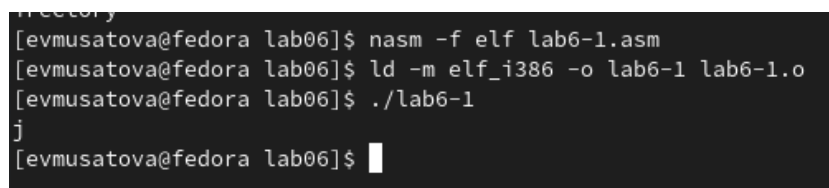
Ввожу в файл lab6-1.asm текст программы из листинга 6.1 (рис. 2.2).

A screenshot of a text editor window titled 'lab6-1.asm' with a path '~/.work/arch-pc/lab06'. The code is as follows:

```
%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. 2.2: Ввод программы

Создаю исполняемый файл и запускаю его для проверки программы и вижу, что выводится символ j (рис. 2.3).

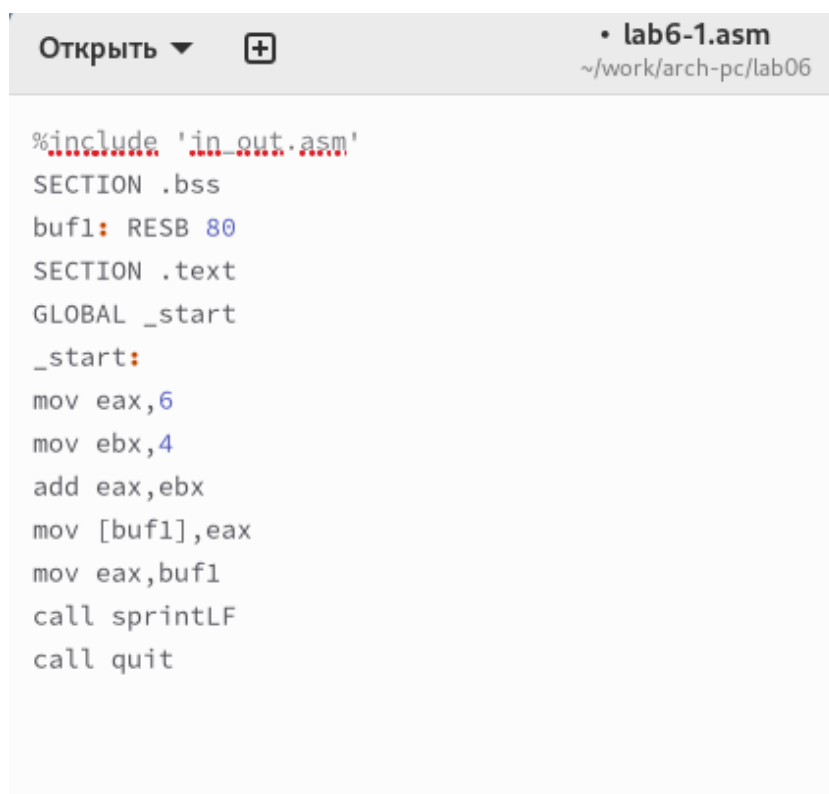
A screenshot of a terminal window showing the following commands and output:

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

Рис. 2.3: Проверка

3

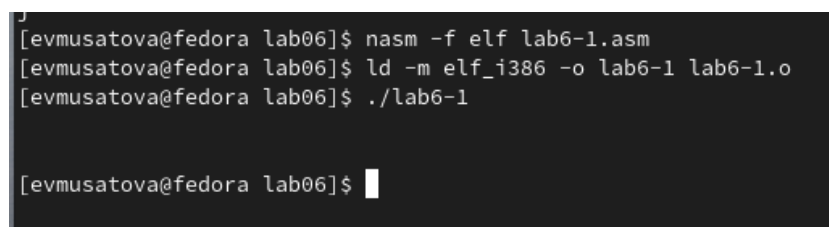
Исправляю текст программы из листинга 6.1, заменяя символы “6” и “4” на цифры 6 и 4 (рис. 2.4).



```
%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: Изменение текста программы

Создаю исполняемый файл и запускаю его (рис. 2.5). Теперь вывелся символ с кодом 10, это символ перевода строки, он не отображается при выводе на экран.



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

[evmusatova@fedora lab06]$
```

Рис. 2.5: Проверка программы

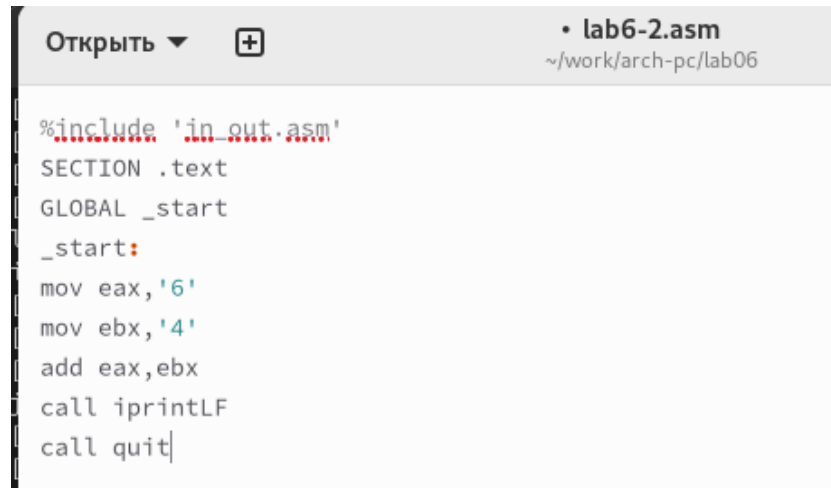
4

Создаю файл lab6-2.asm в каталоге lab06 (рис. 2.6).


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

Рис. 2.6: Создание файла

Ввожу в созданный файл текст программы из листинга 6.2 (рис. 2.7).



```

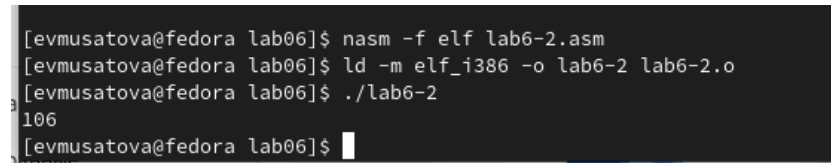
Открыть ▾ + • lab6-2.asm
~/work/arch-pc/lab06

%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: Программа вывода значения регистра eax

Создаю исполняемый файл и запускаю его (рис. 2.8). Теперь выводится число 106, потому что программа позволяет вывести именно число, а не символ, хотя все еще происходит именно сложение кодов символов “6” и “4”.



```

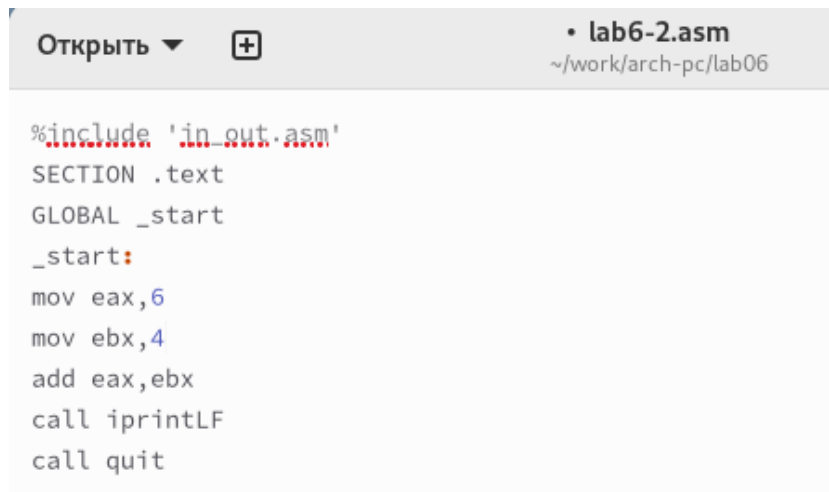
[evmusatova@fedora lab06]$ nasm -f elf lab6-2.asm
[evmusatova@fedora lab06]$ ld -m elf_i386 -o lab6-2 lab6-2.o
[evmusatova@fedora lab06]$ ./lab6-2
106
[evmusatova@fedora lab06]$

```

Рис. 2.8: Проверка программы

5

Аналогично предыдущему примеру изменяю символы на числа (рис. 2.9).



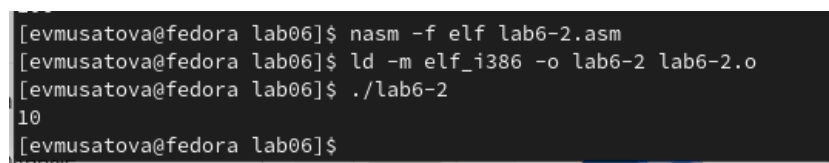
The screenshot shows a text editor window titled "lab6-2.asm" with a path "~/work/arch-pc/lab06". The code inside is as follows:

```
Открыть ▾ + • lab6-2.asm
~/work/arch-pc/lab06

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

Рис. 2.9: Изменение программы

Создаю исполняемый файл и запускаю его (рис. 2.10). Теперь программа складывает не соответствующие символам коды в системе ASCII, а сами числа, поэтому вывод 10.

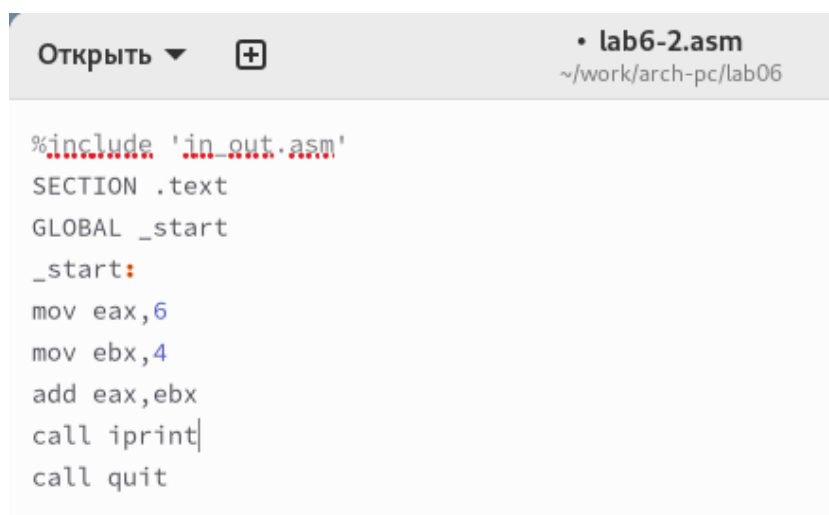


The screenshot shows a terminal window with the following commands and output:

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

Рис. 2.10: Проверка

Заменяю функцию `iprintLF` на `iprint` (рис. 2.11).

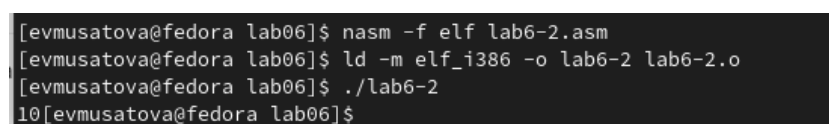


```
Открыть ▾ + lab6-2.asm
~/work/arch-pc/lab06

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

Рис. 2.11: Замена функции

Создаю исполняемый файл и запускаю его (рис. 2.12). Теперь функция `iprint` добавляет к выводу символ переноса строки, в отличие от `iprintLF`.

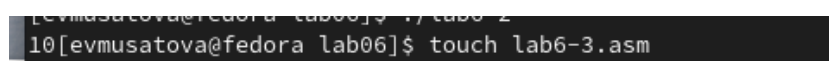


```
[evmusatova@fedora lab06]$ nasm -f elf lab6-2.asm
[evmusatova@fedora lab06]$ ld -m elf_i386 -o lab6-2 lab6-2.o
[evmusatova@fedora lab06]$ ./lab6-2
10[evmusatova@fedora lab06]$
```

Рис. 2.12: Проверка программы

6

Создаю файл `lab6-3` (рис. 2.13).



```
[evmusatova@fedora lab06]$ ./lab6-2
10[evmusatova@fedora lab06]$ touch lab6-3.asm
```

Рис. 2.13: Создание нового файла

Ввожу в текст программы из листинга 6.3 в созданный файл. Затем создаю исполняемый файл и проверяю работу программы (рис. 2.14).

```

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

```

Рис. 2.14: Проверка программы

Изменяю текст программы для вычисления выражения $\text{div}(\text{div}) = (4 \times 6 + 2)/5$ (рис. 2.15).

```

;-----
%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 ; FAX=4
mov ebx,6 ; FBX=6
mul ebx ; FAX=FAX*FBX
add eax,2 ; FAX=FAX+2
xor edx,edx ; обнуляем EDX для корректной работы div
mov ebx,5 ; FBX=5
div ebx ; FAX=FAX/5, EDX=остаток от деления
mov edi,eax ; запись результата вычисления в 'edi'
; ---- Вывод результата на экран
mov eax,div ; вызов подпрограммы печати
call printf ; сообщения 'Результат: '
mov eax,edi ; вызов подпрограммы печати значения

```

Рис. 2.15: Изменение программы

Создаю исполняемый файл и проверяю его работу (рис. 2.16).

```

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

```

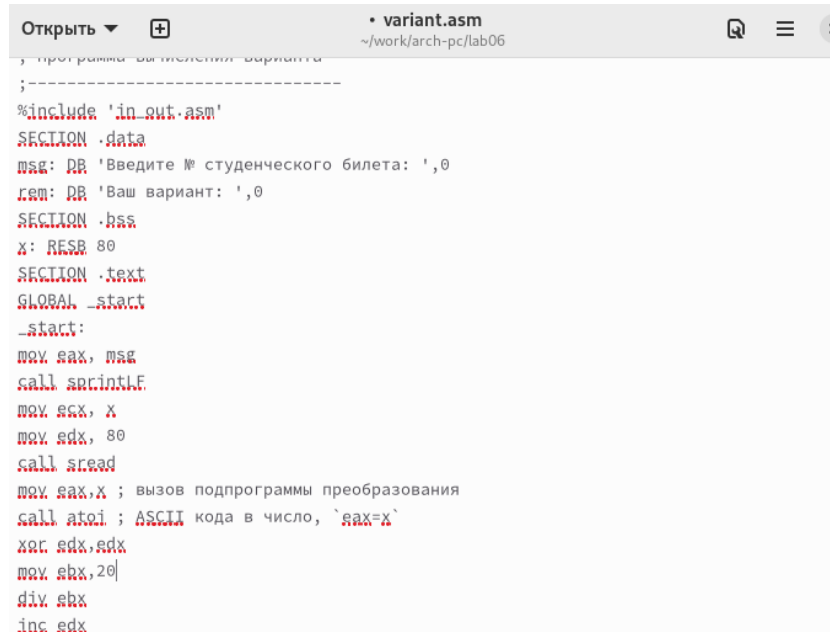
Рис. 2.16: Проверка

Создаю файл variant.asm в каталоге lab06 (рис. 2.17).

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

Рис. 2.17: Создание нового файла

Ввожу текст программы из листинга 6.4 (рис. 2.18).



```
Открыть ▾ + • variant.asm
~/work/arch-pc/lab06
; 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
```

Рис. 2.18: Создание программы

Создаю исполняемый файл и запускаю его, затем ввожу номер студенческого билета и узнаю номер своего варианта (рис. 2.19).

```
[evmusatova@fedora lab06]$ nasm -f elf variant.asm
[evmusatova@fedora lab06]$ ld -m elf_i386 -o variant variant.o
[evmusatova@fedora lab06]$ ./variant
Введите № студенческого билета:
1132236087
Ваш вариант: 8
[evmusatova@fedora lab06]$
```

Рис. 2.19: Проверка работы программы

3 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. Инструкция `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:

```
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`

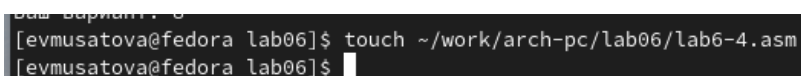
6. Инструкция `inc edx` увеличивает значение регистра `edx` на 1

7. За вывод на экран результатов вычислений отвечают строки:

```
mov eax,edx call iprintLF
```

4 Самостоятельная работа

Создаю файл lab6-4 (рис. 4.1).

A terminal window with a dark background. The prompt is [evmusatova@fedora lab06]\$. The command touch ~/work/arch-pc/lab06/lab6-4.asm is entered. The prompt is repeated on the next line with a cursor.

```
[evmusatova@fedora lab06]$ touch ~/work/arch-pc/lab06/lab6-4.asm
[evmusatova@fedora lab06]$
```

Рис. 4.1: Создание файла

Открываю созданный файл, ввожу в него текст программы для вычисления значения выражения $(11 + x) * 2 - 6$ (рис. 4.2). Это выражение было под вариантом 8, который мне достался.

```

%include 'in_out.asm' ; подключение внешнего файла
SECTION .data ; секция иницированных данных
msg: DB 'Введите значение переменной x: ',0
rem: DB 'Результат: ',0
SECTION .bss ; секция не иницированных данных
x: RESB 80 ; Переменная, значение к-рой будем вводить с клавиатуры, выделенный
размер - 80 байт
SECTION .text ; Код программы
GLOBAL _start ; Начало программы
_start: ; Точка входа в программу
; ---- Вычисление выражения
mov eax, msg ; запись адреса выводимого сообщения в eax
call sprint ; вызов подпрограммы печати сообщения
mov ecx, x ; запись адреса переменной в ecx
mov edx, 80 ; запись длины вводимого значения в edx
call sread ; вызов подпрограммы ввода сообщения
mov eax,x ; вызов подпрограммы преобразования
call atoi ; ASCII кода в число, 'eax=x'
add eax,11; eax = eax+11 = x + 11
mov ebx,2 ; запись значения 2 в регистр ebx
mul ebx; EAX=EAX*EBX = (x+11)*2
add eax,-6; eax = eax-6 = (x+11)*2-6
mov edi,eax ; запись результата вычисления в 'edi'
; ---- Вывод результата на экран
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: Создание программы

Создаю исполняемый файл (рис. 4.3).

```

[evmusatova@fedora lab06]$ nasm -f elf lab6-4.asm
[evmusatova@fedora lab06]$ ld -m elf_i386 -o lab6-4 lab6-4.o

```

Рис. 4.3: Создание исполняемого файла

Запускаю исполняемый файл, ввожу значение - 1, при выводе получаю - 18 (рис. 4.4).

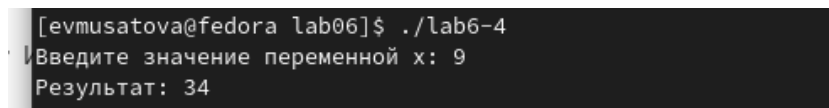
```

[evmusatova@fedora lab06]$ ./lab6-4
Введите значение переменной x: 1
Результат: 18

```

Рис. 4.4: Проверка работы программы

Запускаю исполняемый файл, теперь ввожу значение - 9, при выводе получаю - 34 (рис. 4.5).



```
[evmusatova@fedora lab06]$ ./lab6-4
Введите значение переменной x: 9
Результат: 34
```

Рис. 4.5: Проверка 2

5 Листинг 6.5. Программа для вычисления значения выражения $(11 + x) * 2 - 6$.

%include 'in_out.asm' ; подключение внешнего файла SECTION .data ; секция инициализированных данных msg: DB 'Введите значение переменной x:',0 rem: DB 'Результат:',0 SECTION .bss ; секция не инициализированных данных x: RESB 80 ; Переменная, значение которой будем вводить с клавиатуры, выделенный размер - 80 байт SECTION .text ; Код программы GLOBAL _start ; Начало программы _start: ; Точка входа в программу ; -- Вычисление выражения mov eax, msg ; запись адреса выводимого сообщения в eax call sprint ; вызов подпрограммы печати сообщения mov ecx, x ; запись адреса переменной в ecx mov edx, 80 ; запись длины вводимого значения в edx call sread ; вызов подпрограммы ввода сообщения mov eax, x ; вызов подпрограммы преобразования call atoi ; ASCII кода в число, eax=x add eax, 11; $eax = eax + 11 = x + 11$ mov ebx, 2 ; запись значения 2 в регистр ebx mul ebx; $EAX = EAX * EBX = (x + 11) * 2$ add eax, -6; $eax = eax - 6 = (x + 11) * 2 - 6$ mov edi, eax ; запись результата вычисления в 'edi' ; -- Вывод результата на экран 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

6 Выводы

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