

Отчёт по лабораторной работе №6

дисциплина: архитектура компьютера

Бражко Александра Александровна

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Символьные и численные данные в NASM	8
5	Выполнение арифметических операций в NASM	13
6	Ответы на вопросы	16
7	Выполнение заданий для самостоятельной работы	18
8	Выводы	20
	Список литературы	21

Список иллюстраций

4.1	Создание и переход в каталог, создание файла	8
4.2	Копирование	8
4.3	Ввод программы	9
4.4	Работа программы	9
4.5	Замена	9
4.6	Работа программы	10
4.7	Создание файла	10
4.8	Ввод программы	10
4.9	Работа программы	11
4.10	Замена	11
4.11	Работа программы	11
4.12	Результат после корректировки	12
5.1	Создание файла	13
5.2	Открытие файла и написание программы	13
5.3	Работа команд	14
5.4	Изменяем текст	14
5.5	Работа команд	14
5.6	Создание файла	14
5.7	Открытие файла и написание программы	15
5.8	Работа команд	15
7.1	Создание файла	18
7.2	Открытие файла и написание программы	18
7.3	Работа команды для $x=1$	19
7.4	Работа команды для $x=9$	19

Список таблиц

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

1 Цель работы

Освоение арифметических инструкций языка ассемблера NASM.

2 Задание

1. Символьные и численные данные в NASM
2. Выполнение арифметических операций в NASM
3. Задание 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 систему
/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 про Unix см. в [1–4].

4 Символьные и численные данные в NASM

Создаём каталог для программ лабораторной работы № 6, переходим в него и создаём файл-lab6.asm (рис. 4.1).

```
aabrazhko@dk8n60 ~ $ mkdir ~/work/arch-pc/lab06
aabrazhko@dk8n60 ~ $ cd ~/work/arch-pc/lab06
aabrazhko@dk8n60 ~/work/arch-pc/lab06 $ touch lab6-1.asm
```

Рис. 4.1: Создание и переход в каталог, создание файла

Копируем в текущий каталог файл in_out.asm (рис. 4.2).

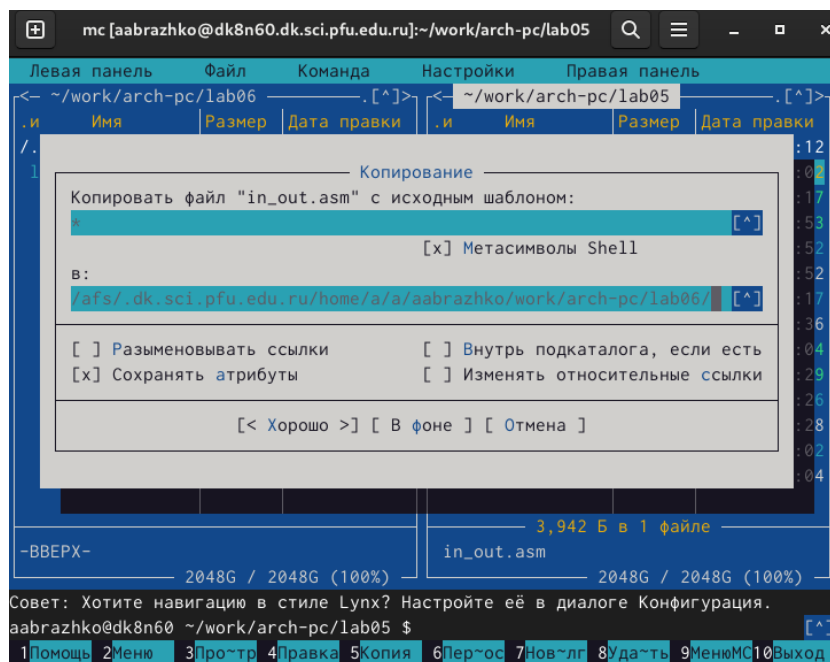


Рис. 4.2: Копирование

Открываем созданный файл lab6-1.asm, вставляем в него программу вывода значения регистра eax (рис. 4.3).

Ввод программы

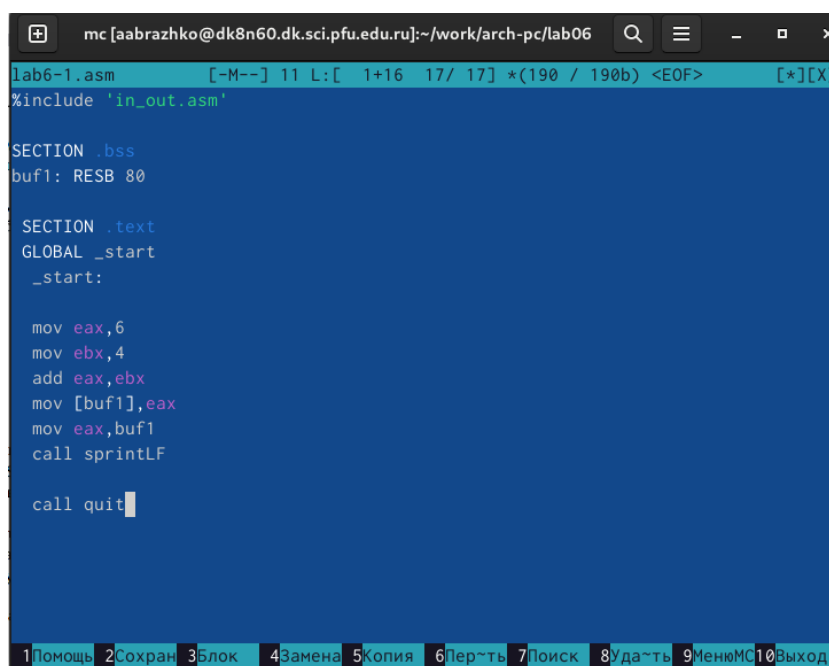
Рис. 4.3: Ввод программы

Создаём и запускаем исполняемый файл. Программа в итоге должна вывести j (рис. 4.4).

```
aabrazhko@dk8n60 ~/work/arch-pc/lab06 $ nasm -f elf lab6-1.asm
aabrazhko@dk8n60 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-1 lab6-1.o
aabrazhko@dk8n60 ~/work/arch-pc/lab06 $ ./lab6-1
j
```

Рис. 4.4: Работа программы

В программе изменяем '4' и '6' на 4 и 6 (рис. 4.5).



```
lab6-1.asm  [-M--] 11 L: [ 1+16 17/ 17] *(190 / 190b) <EOF>  [*][X]
#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
    10
```

Рис. 4.5: Замена

Создаём и запускаем исполняемый файл. Программа в итоге должна вывести символ с кодом 10, но этот символ не отображается при выводе на экран (рис. 4.6).

```

aabrazhko@dk8n60 ~/work/arch-pc/lab06 $ nasm -f elf lab6-1.asm

aabrazhko@dk8n60 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-1 lab6-1.o

aabrazhko@dk8n60 ~/work/arch-pc/lab06 $ ./lab6-1

```

Рис. 4.6: Работа программы

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

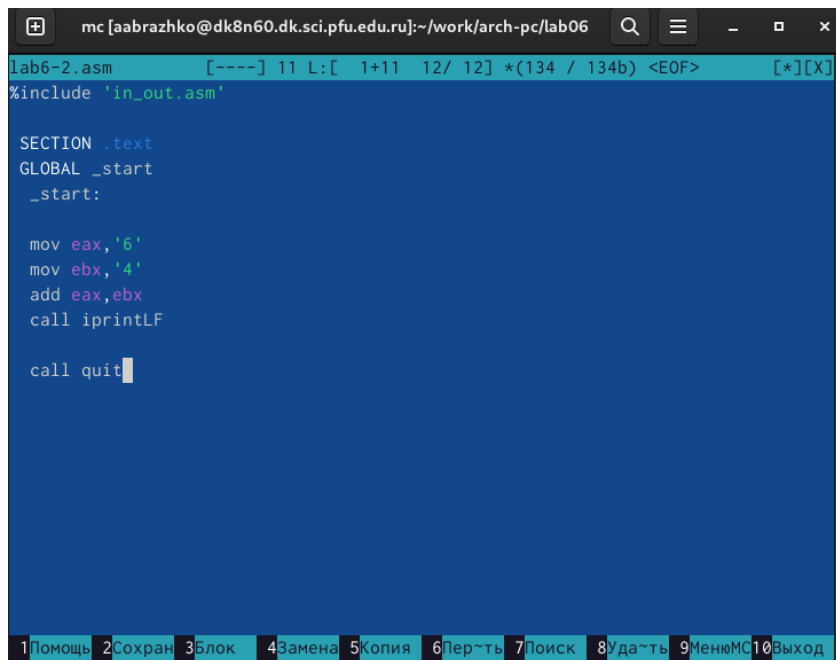
```

aabrazhko@dk8n60 ~/work/arch-pc/lab06 $ touch ~/work/arch-pc/lab06/lab6-2.asm

```

Рис. 4.7: Создание файла

Вводим в него текст программы из листинга 6.2 (рис. 4.8).



```

lab6-2.asm  [----] 11 L: [ 1+11 12/ 12] *(134 / 134b) <EOF>  [*][X]
#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 программы

Создаём и запускаем исполняемый файл. Программа в итоге должна вывести 106 (рис. 4.9).

```

aabrazhko@dk8n60 ~/work/arch-pc/lab06 $ nasm -f elf lab6-2.asm
aabrazhko@dk8n60 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-2 lab6-2.o
aabrazhko@dk8n60 ~/work/arch-pc/lab06 $ ./lab6-2
106

```

Рис. 4.9: Работа программы

В программе изменяем '4' и '6' на 4 и 6 (рис. 4.10).

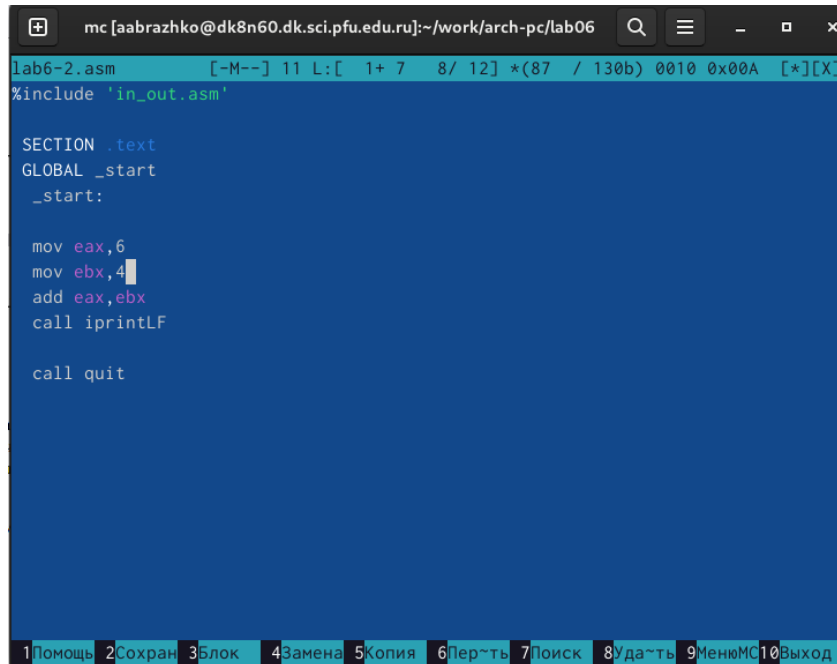


Рис. 4.10: Замена

Создаём и запускаем исполняемый файл. Программа в итоге должна вывести 10 (рис. 4.11).

```

aabrazhko@dk8n60 ~/work/arch-pc/lab06 $ nasm -f elf lab6-2.asm
aabrazhko@dk8n60 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-2 lab6-2.o
aabrazhko@dk8n60 ~/work/arch-pc/lab06 $ ./lab6-2
10

```

Рис. 4.11: Работа программы

Заменяем функцию iprintLF на iprint. Создаём исполняемый файл и запускаем его. Разницей между iprintLF и iprint является количество строк вывода (рис. 4.12).

```
aabrazhko@dk8n60 ~/work/arch-pc/lab06 $ nasm -f elf lab6-2.asm
aabrazhko@dk8n60 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-2 lab6-2.o
aabrazhko@dk8n60 ~/work/arch-pc/lab06 $ ./lab6-2
10aabrazhko@dk8n60 ~/work/arch-pc/lab06 $
```

Рис. 4.12: Результат после корректировки

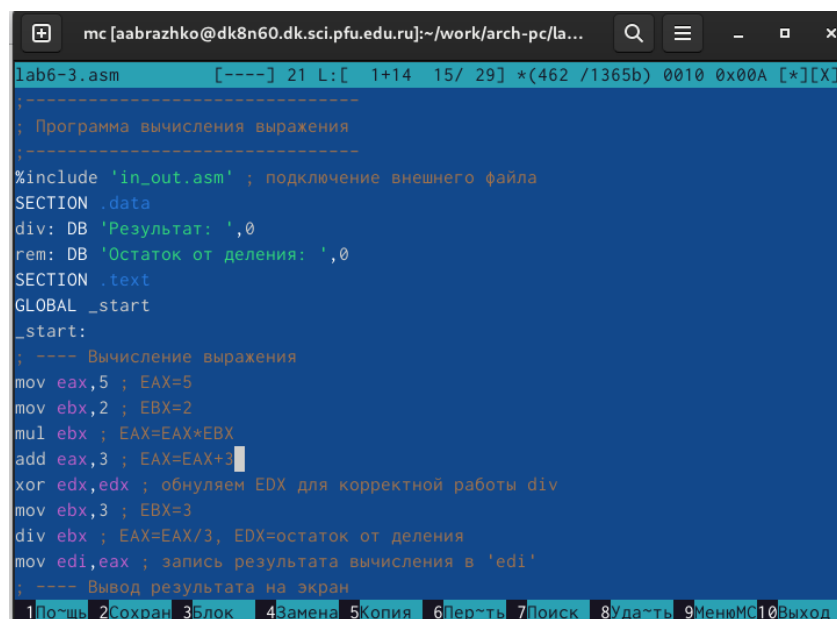
5 Выполнение арифметических операций в NASM

Создаём файл lab6-3.asm в каталоге ~/work/arch-pc/lab06 (рис. 5.1).

```
aabrazhko@dk8n60 ~/work/arch-pc/lab06 $ touch ~/work/arch-pc/lab06/lab6-3.asm
```

Рис. 5.1: Создание файла

Открываем созданный файл lab6-3.asm, вставляем в него текст программы из листинга 6.3 (рис. 5.2).



```
lab6-3.asm [----] 21 L:[ 1+14 15/ 29] *(462 /1365b) 0010 0x00A [*][X]
;-----
; 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
1Помощь 2Сохран 3Блок 4Замена 5Копия 6Пер-ть 7Поиск 8Уда-ть 9МенюМС 10Выход
```

Рис. 5.2: Открытие файла и написание программы

Создаём и запускаем исполняемый файл (рис. 5.3).

```

aabrazhko@dk8n60 ~/work/arch-pc/lab06 $ nasm -f elf lab6-3.asm
aabrazhko@dk8n60 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-3 lab6-3.o
aabrazhko@dk8n60 ~/work/arch-pc/lab06 $ ./lab6-3
Результат: 4
Остаток от деления: 1

```

Рис. 5.3: Работа команд

Изменяем текст программы для вычисления выражения $\square(\square) = (4 \square 6 + 2)/5$ (рис. 5.4).

```

lab6-3.asm      [-M--] 50 L:[ 1+18 19/ 29] *(698 /1365b) 0010 0x00A [*][X]
;-----
; 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=5
mov ebx,6 ; EBX=2
mul ebx ; EAX=EAX*EBX
add eax,2 ; EAX=EAX+3
xor edx,edx ; обнуляем EDX для корректной работы div
mov ebx,5 ; EBX=3
div ebx ; EAX=EAX/3, EDX=остаток от деления
mov edi,eax ; запись результата вычисления в 'edi'
; ---- Вывод результата на экран
1Поиск 2Сохран 3Блок 4Замена 5Копия 6Перейти 7Поиск 8Удалить 9Меню 10Выход

```

Рис. 5.4: Изменяем текст

Создаём и запускаем исполняемый файл (рис. 5.5).

```

aabrazhko@dk8n60 ~/work/arch-pc/lab06 $ nasm -f elf lab6-3.asm
aabrazhko@dk8n60 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-3 lab6-3.o
aabrazhko@dk8n60 ~/work/arch-pc/lab06 $ ./lab6-3
Результат: 5
Остаток от деления: 1

```

Рис. 5.5: Работа команд

Создаём файл variant.asm в каталоге ~/work/arch-pc/lab06 (рис. 5.6).

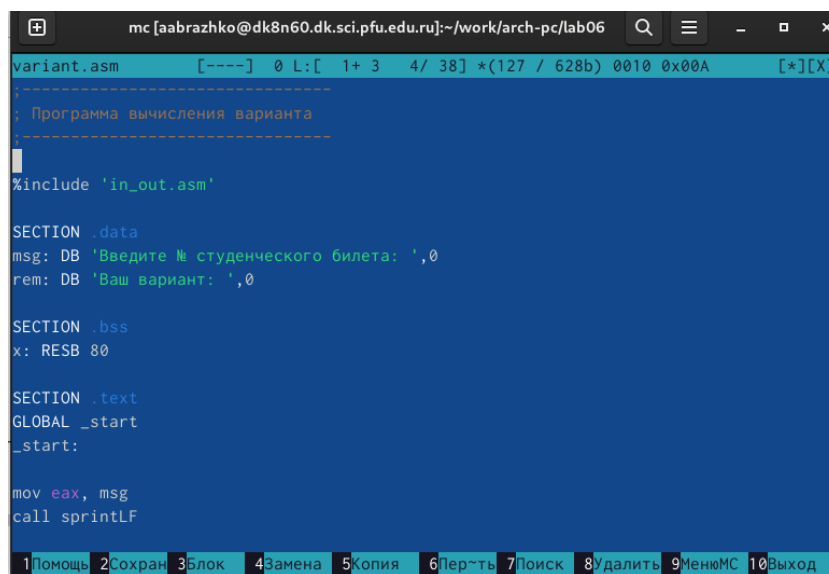
```

aabrazhko@dk8n60 ~/work/arch-pc/lab06 $ touch ~/work/arch-pc/lab06/variant.asm

```

Рис. 5.6: Создание файла

Открываем созданный файл `variant.asm`, вставляем в него текст программы из листинга 6.4 (рис. 5.7).



```
variant.asm [----] 0 L: [ 1+ 3 4/ 38] *(127 / 628b) 0010 0x00A [*][X]
;-----
; Программа вычисления варианта
;-----
#include 'in_out.asm'

SECTION .data
msg: DB 'Введите № студенческого билета: ',0
rem: DB 'Ваш вариант: ',0

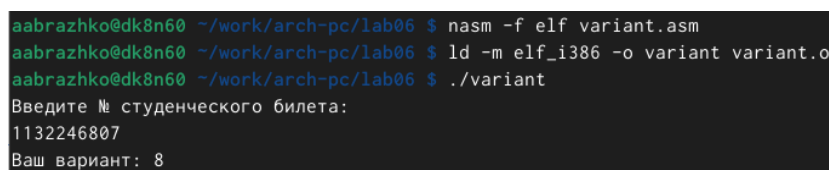
SECTION .bss
x: RESB 80

SECTION .text
GLOBAL _start
_start:

mov eax, msg
call sprintfLF
```

Рис. 5.7: Открытие файла и написание программы

Создаём и запускаем исполняемый файл (рис. 5.8).



```
aabrazhko@dk8n60 ~/work/arch-pc/lab06 $ nasm -f elf variant.asm
aabrazhko@dk8n60 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o variant variant.o
aabrazhko@dk8n60 ~/work/arch-pc/lab06 $ ./variant
Введите № студенческого билета:
1132246807
Ваш вариант: 8
```

Рис. 5.8: Работа команд

6 Ответы на вопросы

- 1) Какие строки листинга 6.4 отвечают за вывод на экран сообщения ‘Ваш вариант:’?

```
mov eax,rem call sprint
```

- 2) Для чего используются следующие инструкции? `mov ecx, x` `mov edx, 80` `call sread`

инструкция `mov ecx, x` используется для того, 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) Для чего используется инструкция “`call atoi`”?

`Call atoi` используется для вызова подпрограммы из внешнего файла, которая преобразует `ascii`-код символа в целое число и записывает результат в регистр `eax`.

- 4) Какие строки листинга 6.4 отвечают за вычисления варианта?

```
xor edx,edx mov ebx,20 div ebx inc edx
```

- 5) В какой регистр записывается остаток от деления при выполнении инструкции “`div ebx`”?

При выполнении инструкции `div ebx` остаток от деления записывается в регистр `edx`.

6) Для чего используется инструкция `inc edx`?

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

7) Какие строки листинга 6.4 отвечают за вывод на экран результата вычислений?

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

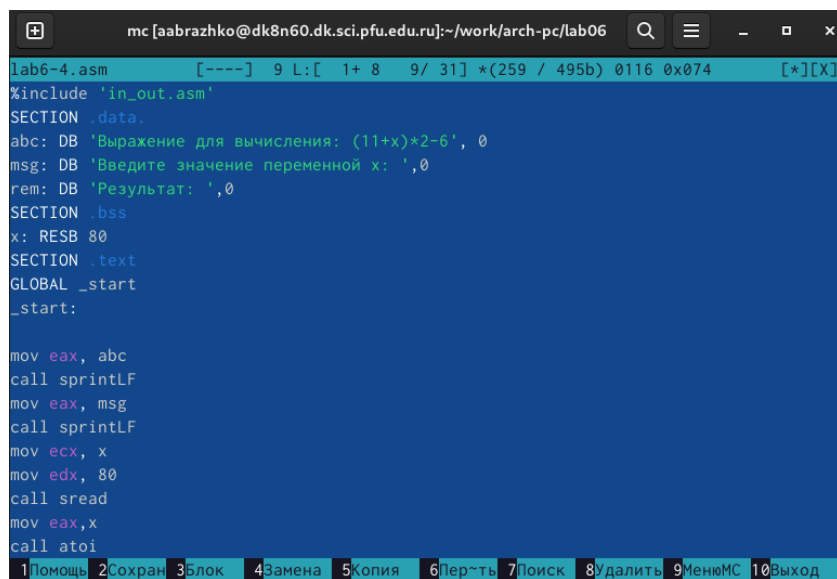
7 Выполнение заданий для самостоятельной работы

Создаём файл lab6-4.asm в каталоге ~/work/arch-pc/lab06 (рис. 7.1).

```
aabrazhko@dk8n60 ~/work/arch-pc/lab06 $ touch ~/work/arch-pc/lab06/lab6-4.asm
```

Рис. 7.1: Создание файла

Открываем созданный файл lab6-4.asm и вводим в него текст программы для вычисления значения выражения для 8 варианта: $(11+x)*2-6$ (рис. 7.2).



```
lab6-4.asm [----] 9 L: [ 1+ 8 9/ 31] *(259 / 495b) 0116 0x074 [*][X]
#include 'in_out.asm'
SECTION .data
abc: DB 'Выражение для вычисления: (11+x)*2-6', 0
msg: DB 'Введите значение переменной x: ', 0
rem: DB 'Результат: ', 0
SECTION .bss
x: RESB 80
SECTION .text
GLOBAL _start
_start:

mov eax, abc
call sprintf
mov eax, msg
call sprintf
mov ecx, x
mov edx, 80
call sread
mov eax, x
call atoi
```

Рис. 7.2: Открытие файла и написание программы

Создаём и запускаем исполняемый файл для $x=1$ и $x=9$ (рис. 7.3, рис. 7.4).

```
aabrazhko@dk8n60 ~/work/arch-pc/lab06 $ nasm -f elf lab6-4.asm
aabrazhko@dk8n60 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-4 lab6-4.o
aabrazhko@dk8n60 ~/work/arch-pc/lab06 $ ./lab6-4
Выражение для вычисления: (11+x)*2-6
Введите значение переменной x:
1
Результат:
18
```

Рис. 7.3: Работа команды для $x=1$

```
aabrazhko@dk8n60 ~/work/arch-pc/lab06 $ nasm -f elf lab6-4.asm
aabrazhko@dk8n60 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-4 lab6-4.o
aabrazhko@dk8n60 ~/work/arch-pc/lab06 $ ./lab6-4
Выражение для вычисления: (11+x)*2-6
Введите значение переменной x:
9
Результат:
34
```

Рис. 7.4: Работа команды для $x=9$

8 Выводы

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

Список литературы

1. Таненбаум Э., Бос Х. Современные операционные системы. 4-е изд. СПб.: Питер, 2015. 1120 с.
2. Robbins A. Bash Pocket Reference. O'Reilly Media, 2016. 156 с.
3. Zarrelli G. Mastering Bash. Packt Publishing, 2017. 502 с.
4. Newham C. Learning the bash Shell: Unix Shell Programming. O'Reilly Media, 2005. 354 с.