

Отчёт по лабораторной работе №6

Архитектура компьютера НММбд-03-24

Туева Анастасия Юрьевна

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
4	Выполнение самостоятельной работы	19
5	Выводы	22

Список иллюстраций

3.1	Создание файла	7
3.2	Редактирование файла	8
3.3	Запуск исполняемого файла	8
3.4	Редактирование файла	9
3.5	Запуск исполняемого файла	9
3.6	Редактирование файла	10
3.7	Запуск исполняемого файла	10
3.8	Редактирование файла	11
3.9	Запуск исполняемого файла	11
3.10	Редактирование файла	12
3.11	Запуск исполняемого файла	12
3.12	Создание файла	12
3.13	Редактирование файла	13
3.14	Запуск исполняемого файла	14
3.15	Изменение программы	15
3.16	Запуск исполняемого файла	16
3.17	Создание файла	16
3.18	Редактирование файла	17
3.19	Запуск исполняемого файла	18
4.1	Создание файла	19
4.2	Написание программы	20
4.3	Запуск исполняемого файла	21

Список таблиц

1 Цель работы

Цель данной лабораторной работы - освоить арифметические инструкции 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

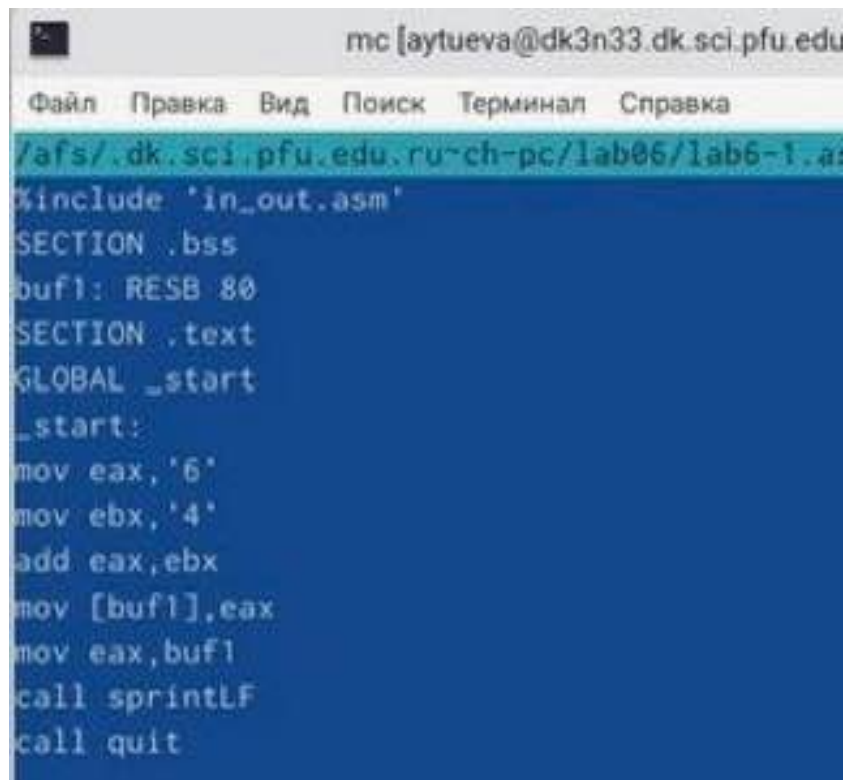
1. Символьные и численные данные в NASM

С помощью команды “touch” создаю файл “lab7-1.asm” (рис. 3.1).

```
aytueva@dk3n33 ~ $ mkdir ~/work/arch-pc/lab06  
aytueva@dk3n33 ~ $ cd ~/work/arch-pc/lab06  
aytueva@dk3n33 ~/work/arch-pc/lab06 $ touch lab6-1.asm  
aytueva@dk3n33 ~/work/arch-pc/lab06 $
```

Рис. 3.1: Создание файла

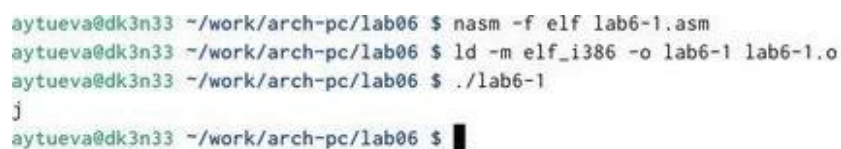
Открываем созданный файл lab7-1.asm, вставляем в него программу вывода значения регистра “eax” (рис. 3.2).



```
mc [aytueva@dk3n33.dk.sci.pfu.edu
Файл  Правка  Вид  Поиск  Терминал  Справка
/afs/.dk.sci.pfu.edu.ru~ch-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. 3.2: Редактирование файла

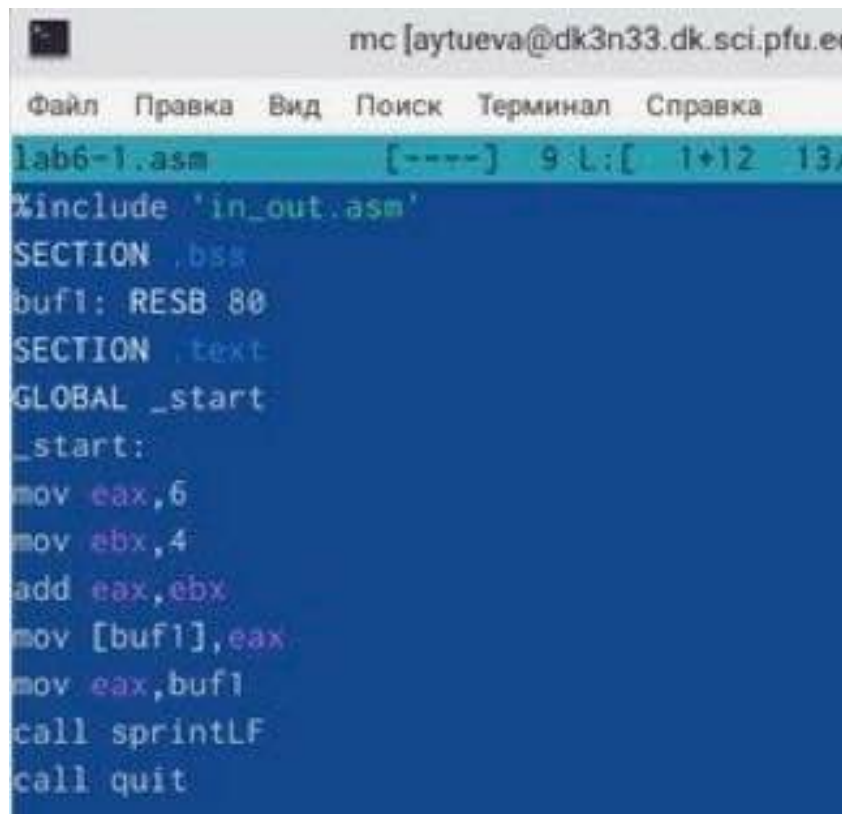
Создаем исполняемый файл программы и запускаем его (рис. 3.3). Программа выводит символ j, потому что программа вывела символ, соответствующий по системе ASCII сумме двоичных кодов символов 4 и 6.



```
aytueva@dk3n33 ~/work/arch-pc/lab06 $ nasm -f elf lab6-1.asm
aytueva@dk3n33 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-1 lab6-1.o
aytueva@dk3n33 ~/work/arch-pc/lab06 $ ./lab6-1
j
aytueva@dk3n33 ~/work/arch-pc/lab06 $
```

Рис. 3.3: Запуск исполняемого файла

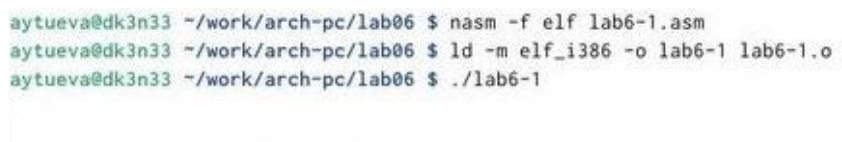
Изменяем в тексте программы символы “6” и “4” на цифры 6 и 4 (рис. 3.4).

A screenshot of a text editor window titled 'mc [aytueva@dk3n33.dk.sci.pfu.e...'. The menu bar includes 'Файл', 'Правка', 'Вид', 'Поиск', 'Терминал', and 'Справка'. The file 'lab6-1.asm' is open, showing assembly code. The code includes a header file, defines a buffer, and contains a main function that prints the number 6 to the console.

```
lab6-1.asm [----] 9: L: [ 1+12 13]
%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: Редактирование файла

Создаем новый исполняемый файл программы и запускаем его (рис. 3.5). Теперь вывелся символ с кодом 10, это символ перевода строки. Этот символ не отображается при выводе на экран.

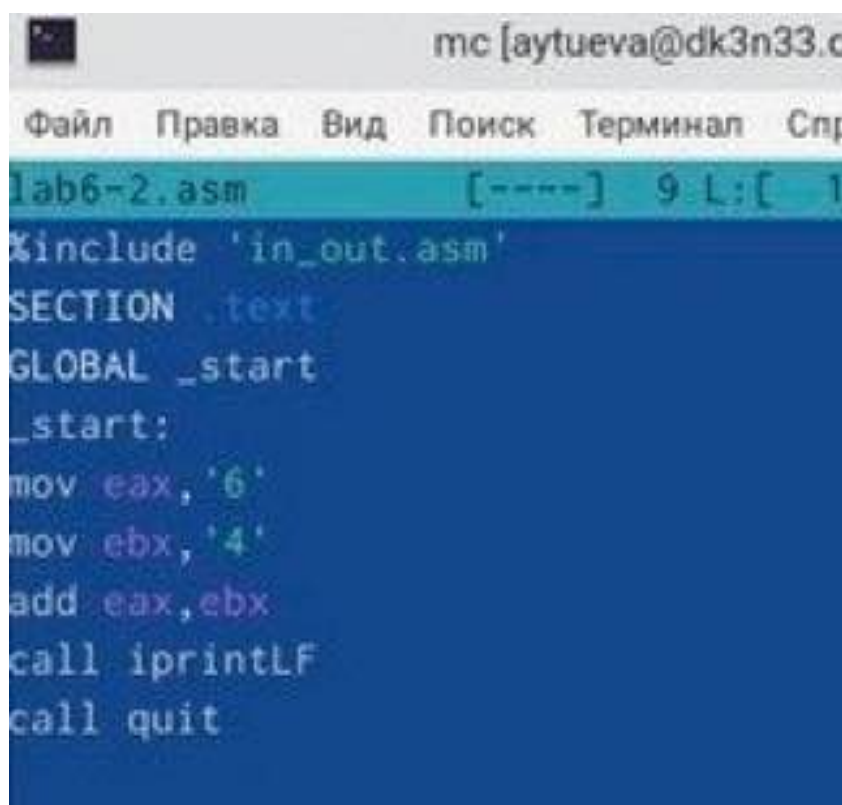
A screenshot of a terminal window showing the compilation and execution of the assembly file. The user runs 'nasm' to compile 'lab6-1.asm' into 'lab6-1.o', then 'ld' to link it into an executable 'lab6-1', and finally runs './lab6-1' which outputs the number 6.

```
aytueva@dk3n33 ~/work/arch-pc/lab06 $ nasm -f elf lab6-1.asm
aytueva@dk3n33 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-1 lab6-1.o
aytueva@dk3n33 ~/work/arch-pc/lab06 $ ./lab6-1
6
```

Рис. 3.5: Запуск исполняемого файла

Создаем новый файл lab7-2.asm с помощью команды “touch”.

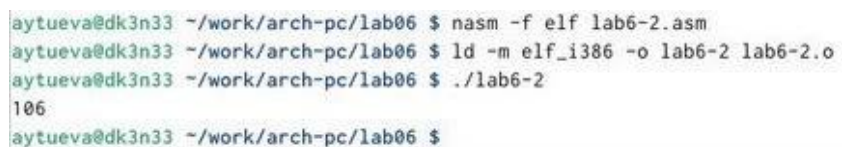
Вводим в файл текст другой программы для вывода значения регистра “eax” (рис. 3.6).

A screenshot of a text editor window titled 'mc [aytueva@dk3n33.c'. The menu bar includes 'Файл', 'Правка', 'Вид', 'Поиск', 'Терминал', and 'Спр'. The file 'lab6-2.asm' is open, showing assembly code. The code includes 'in_out.asm', defines a text section, and sets up a program entry point. It moves the values 6 and 4 into registers, adds them, and prints the result.

```
lab6-2.asm [-----] 9 L:[ 1
#include 'in_out.asm'
SECTION .text
GLOBAL _start
_start:
mov eax, '6'
mov ebx, '4'
add eax, ebx
call iprintLF
call quit
```

Рис. 3.6: Редактирование файла

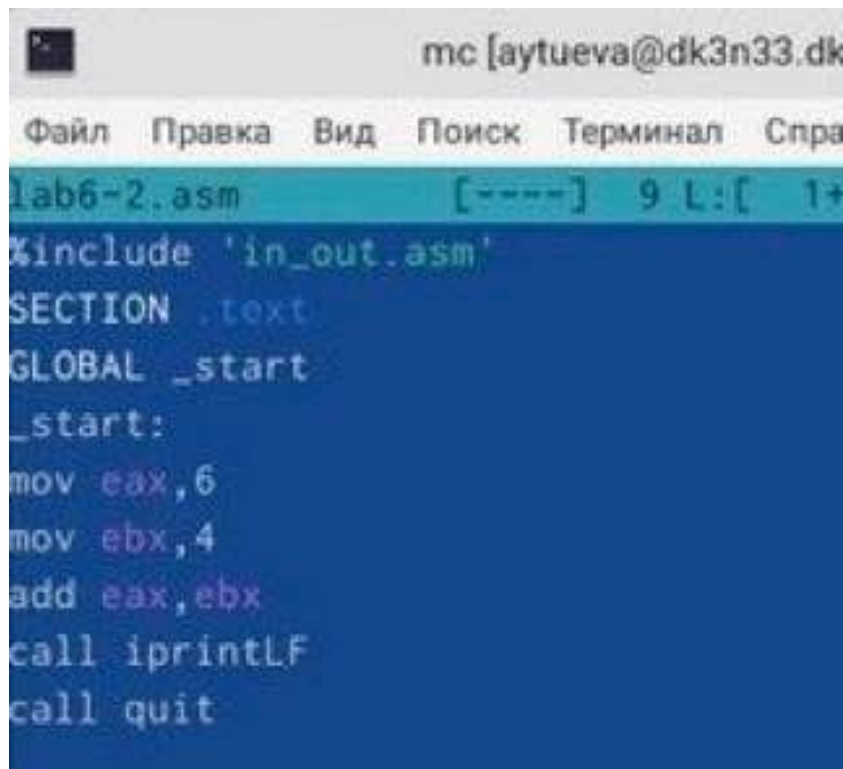
Создаем и запускаем исполняемый файл “lab7-2” (рис. 3.7). Теперь выводится число 106.

A screenshot of a terminal window showing the compilation and execution of the assembly file. The user runs 'nasm' to compile 'lab6-2.asm' into 'lab6-2.o', then 'ld' to link it into an executable 'lab6-2', and finally runs the executable, which outputs '106'.

```
aytueva@dk3n33 ~/work/arch-pc/lab06 $ nasm -f elf lab6-2.asm
aytueva@dk3n33 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-2 lab6-2.o
aytueva@dk3n33 ~/work/arch-pc/lab06 $ ./lab6-2
106
aytueva@dk3n33 ~/work/arch-pc/lab06 $
```

Рис. 3.7: Запуск исполняемого файла

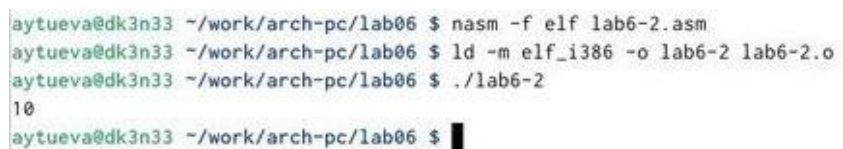
Заменяем в тексте программы в файле “lab7-2.asm” символы “6” и “4” на числа 6 и 4 (рис. 3.8).

A screenshot of a text editor window titled 'mc [aytueva@dk3n33.dk]'. The menu bar includes 'Файл', 'Правка', 'Вид', 'Поиск', 'Терминал', and 'Спра'. The file 'lab6-2.asm' is open, showing assembly code. The code includes a directive to include 'in_out.asm', a section declaration '.text', and a global symbol '_start'. The main code block starts with '_start:' followed by instructions: 'mov eax,6', 'mov ebx,4', 'add eax,ebx', 'call iprintLF', and 'call quit'.

```
mc [aytueva@dk3n33.dk]
Файл  Правка  Вид  Поиск  Терминал  Спра
lab6-2.asm  [-----]  9  L:[  1+
%include 'in_out.asm'
SECTION .text
GLOBAL _start
_start:
mov eax,6
mov ebx,4
add eax,ebx
call iprintLF
call quit
```

Рис. 3.8: Редактирование файла

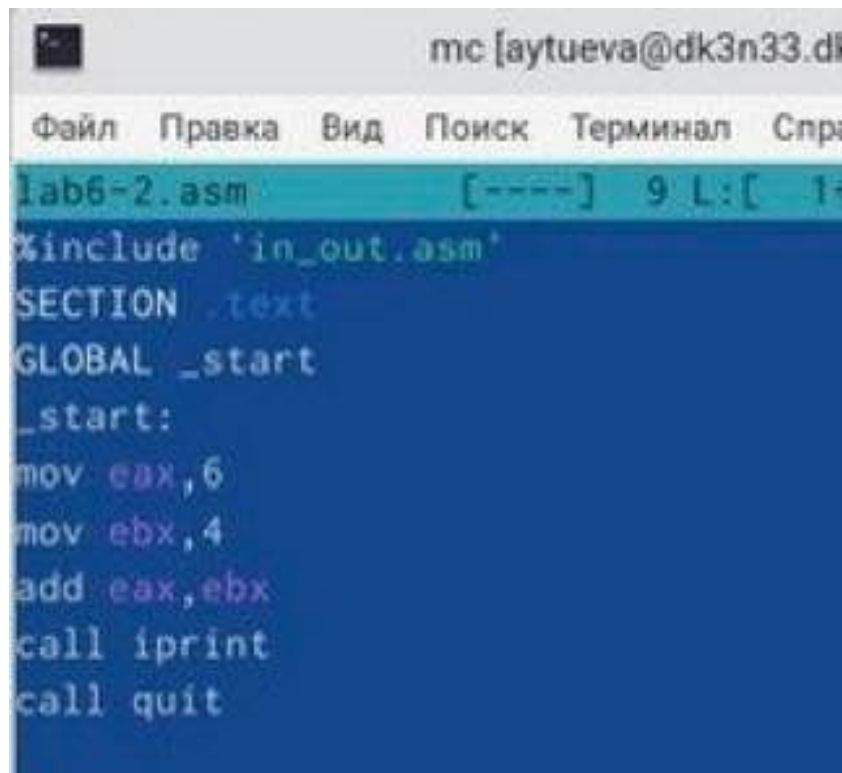
Создаем и запускаем новый исполняемый файл (рис. 3.9).. Теперь программа складывает не соответствующие символам коды в системе ASCII, а сами числа, поэтому выводится 10.

A screenshot of a terminal window showing the compilation and execution of the assembly file. The user runs 'nasm -f elf lab6-2.asm', then 'ld -m elf_i386 -o lab6-2 lab6-2.o', and finally './lab6-2'. The output of the program is '10'.

```
aytueva@dk3n33 ~/work/arch-pc/lab06 $ nasm -f elf lab6-2.asm
aytueva@dk3n33 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-2 lab6-2.o
aytueva@dk3n33 ~/work/arch-pc/lab06 $ ./lab6-2
10
aytueva@dk3n33 ~/work/arch-pc/lab06 $
```

Рис. 3.9: Запуск исполняемого файла

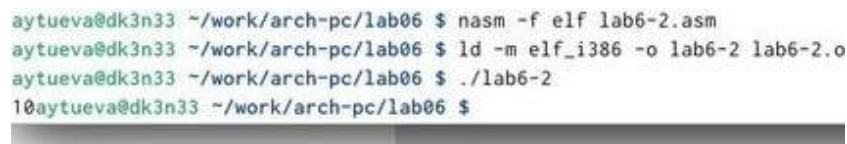
Заменяем в тексте программы функцию “iprintLF” на “iprint” (рис. 3.10).



```
mc [aytueva@dk3n33.dl
Файл Правка Вид Поиск Терминал Спр
lab6-2.asm [----] 9 L:[ 1+
%include 'in_out.asm'
SECTION .text
GLOBAL _start
_start:
mov eax,6
mov ebx,4
add eax,ebx
call iprint
call quit
```

Рис. 3.10: Редактирование файла

Создаем и запускаем новый исполняемый файл (рис. 3.11). Вывод не изменился.

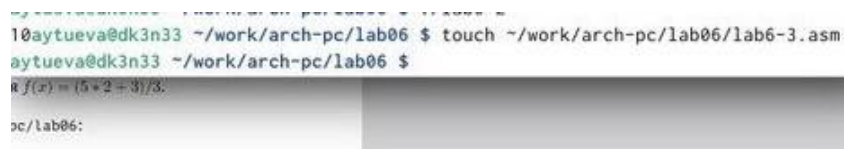


```
aytueva@dk3n33 ~/work/arch-pc/lab06 $ nasm -f elf lab6-2.asm
aytueva@dk3n33 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-2 lab6-2.o
aytueva@dk3n33 ~/work/arch-pc/lab06 $ ./lab6-2
10aytueva@dk3n33 ~/work/arch-pc/lab06 $
```

Рис. 3.11: Запуск исполняемого файла

2. Выполнение арифметических операций в NASM

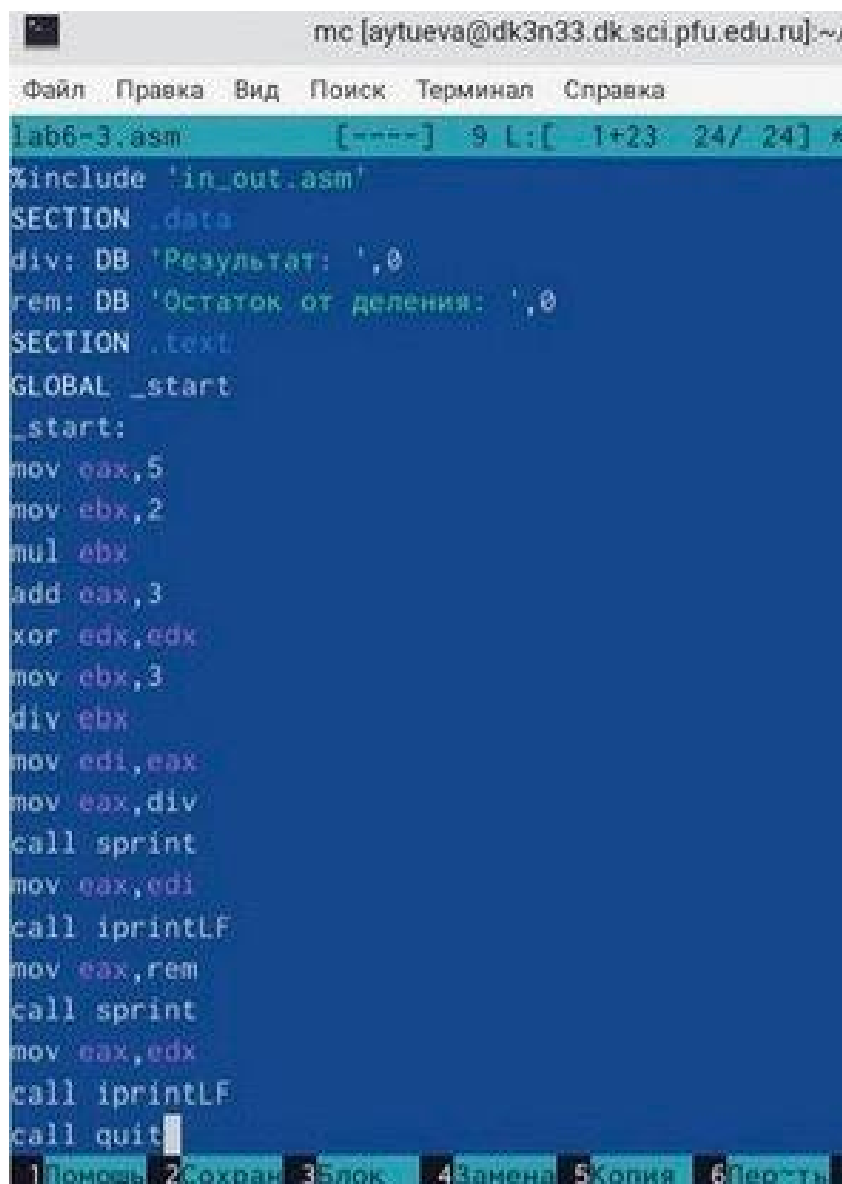
Создаем файл “lab7-3.asm” с помощью команды “touch” (рис. 3.12).



```
10aytueva@dk3n33 ~/work/arch-pc/lab06 $ touch ~/work/arch-pc/lab06/lab6-3.asm
aytueva@dk3n33 ~/work/arch-pc/lab06 $
# f(x) = (5 * 2 + 3) / 3.
pc/lab06:
```

Рис. 3.12: Создание файла

Вводим в созданный файл текст программы для вычисления значения выражения $f(x) = (5 * 2 + 3)/3$ (рис. 3.13).



```
mc [aytueva@dk3n33.dk.sci.pfu.edu.ru]:~/
Файл Правка Вид Поиск Терминал Справка
lab6-3.asm [----] 9 L: [ 1+23 24/ 24] *
#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. 3.13: Редактирование файла

Создаем исполняемый файл и запускаю его (рис. 3.14).

```
aytueva@dk3n33 ~/work/arch-pc/lab06 $ nasm -f elf lab6-3.asm
aytueva@dk3n33 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-3 lab6-3.o
aytueva@dk3n33 ~/work/arch-pc/lab06 $ ./lab6-3
Результат: 4
Остаток от деления: 1
```

Рис. 3.14: Запуск исполняемого файла

Изменяем программу так, чтобы она вычисляла значение выражения “ $f(x) = (4 * 6 + 2)/5$ ” (рис. 3.15).



```
mc [aytueva@dk3n33.dk.scl]
Файл Правка Вид Поиск Терминал Справка
lab6-3.asm [----] 9 L: [ 1+23
#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. 3.15: Изменение программы

Создаем и запускаем новый исполняемый файл (рис. 3.16).

```
aytueva@dk3n33 ~/work/arch-pc/lab06 $ nasm -f elf lab6-3.asm
aytueva@dk3n33 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-3 lab6-3.o
aytueva@dk3n33 ~/work/arch-pc/lab06 $ ./lab6-3
Результат: 5
Остаток от деления: 1
```

Рис. 3.16: Запуск исполняемого файла

Создаем файл “variant.asm” с помощью команды “touch” (рис. 3.17).

```
Остаток от деления: 1
aytueva@dk3n33 ~/work/arch-pc/lab06 $ touch ~/work/arch-pc/lab06/variant.asm
aytueva@dk3n33 ~/work/arch-pc/lab06 $
```

Рис. 3.17: Создание файла

Вводим в файл текст программы для вычисления варианта задания по номеру студенческого билета (рис. 3.18).



```
mc [aytueva@dk3n33.dk.sci.pfu.edu.ru]:~/
Файл Правка Вид Поиск Терминал Справка
variant.asm [----] 9 L: [ 1+24 25/ 25]
#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
call atoi
xor edx, edx
mov ebx, 20
div ebx
inc edx
mov eax, rem
call sprintf
mov eax, edx
call iprintLF
call quit
```

Рис. 3.18: Редактирование файла

Создаем и запускаем исполняемый файл (рис. 3.19). Вводим номер своего студенческого билета с клавиатуры, программа выводит, что мой вариант - 11.

```

aytueva@dk3n33 ~/work/arch-pc/lab06 $ nasm -f elf variant.asm
aytueva@dk3n33 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o variant variant.o
aytueva@dk3n33 ~/work/arch-pc/lab06 $ ./variant
Введите № студенческого билета:
1132242910
Ваш вариант: 11

```

Рис. 3.19: Запуск исполняемого файла

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:

```
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 Выполнение самостоятельной работы

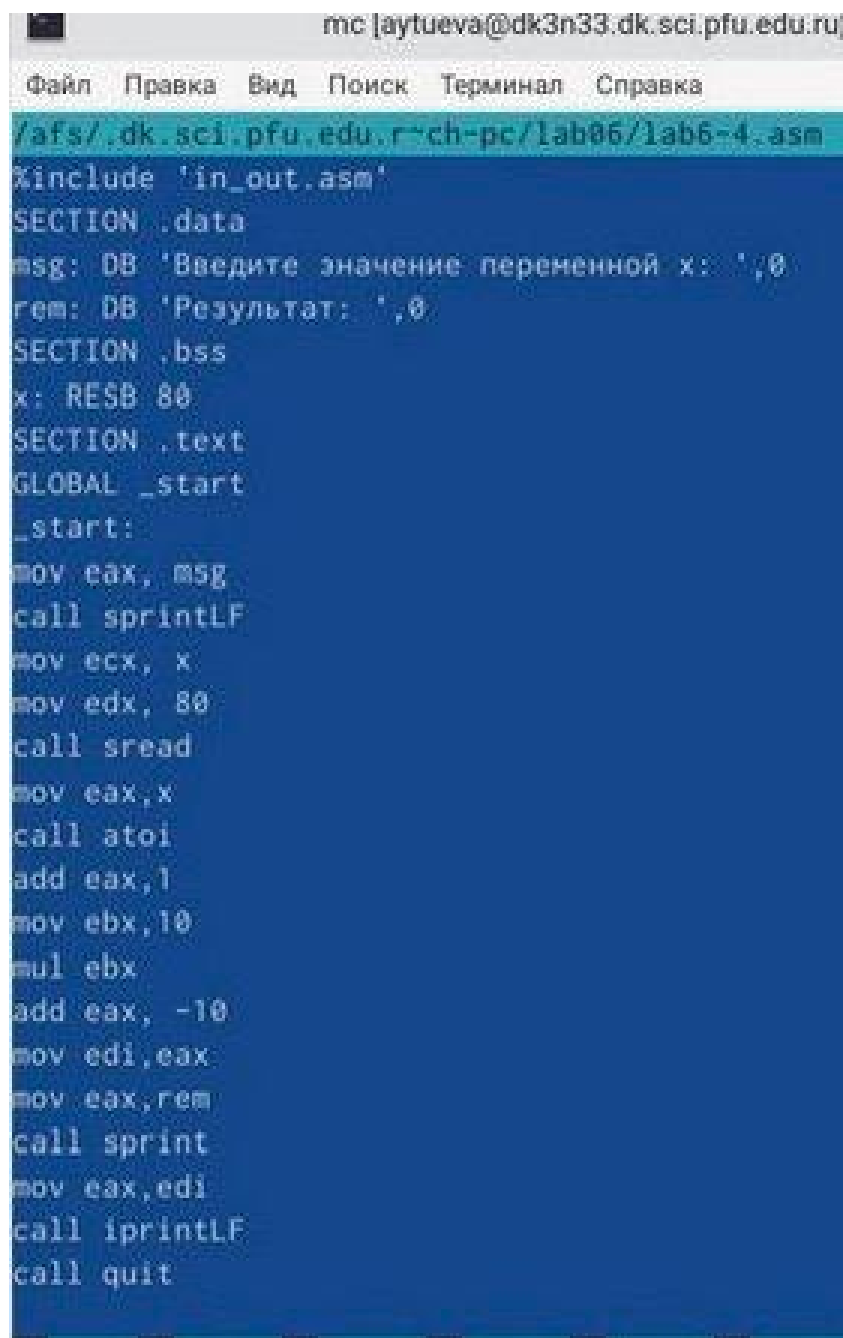
Создаем файл “lab7-4.asm” с помощью команды “touch” (рис. 4.1).

A screenshot of a terminal window. The first line shows a prompt "ваш вариант: 11". The second line shows a user "aytueva@dk3n33" at a shell prompt "\$" in the directory "~/work/arch-pc/lab06", typing the command "touch ~/work/arch-pc/lab06/lab6-4.asm". The third line shows the same user and prompt, with the command completed.

```
ваш вариант: 11
aytueva@dk3n33 ~/work/arch-pc/lab06 $ touch ~/work/arch-pc/lab06/lab6-4.asm
aytueva@dk3n33 ~/work/arch-pc/lab06 $
```

Рис. 4.1: Создание файла

Открываем созданный файл для редактирования, вводим в него текст программы для вычисления значения выражения “ $10 * (x + 1) - 10$ ” (рис. 4.2). Это выражение было под вариантом 11.



```
mc | aytueva@dk3n33.dk.sci.pfu.edu.ru
Файл  Правка  Вид  Поиск  Терминал  Справка
/afs/.dk.sci.pfu.edu.r~ch-pc/lab06/lab6-4.asm
%include 'in_out.asm'
SECTION .data
msg: DB 'Введите значение переменной x: ',0
rem: DB 'Результат: ',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
call atoi
add eax, 1
mov ebx, 10
mul ebx
add eax, -10
mov edi, eax
mov eax, rem
call sprintf
mov eax, edi
call iprintLF
call quit
```

Рис. 4.2: Написание программы

Создаем и запускаем исполняемый файл (рис. 4.3). При вводе значения 1, вывод - 10, а при вводе значения 7, вывод - 70.

```
aytueva@dk3n33 ~/work/arch-pc/lab06 $ nasm -f elf lab6-4.asm
aytueva@dk3n33 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-4 lab6-4.o
aytueva@dk3n33 ~/work/arch-pc/lab06 $ ./lab6-4
Введите значение переменной x:
1
Результат: 10
aytueva@dk3n33 ~/work/arch-pc/lab06 $ nasm -f elf lab6-4.asm
aytueva@dk3n33 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-4 lab6-4.o
aytueva@dk3n33 ~/work/arch-pc/lab06 $ ./lab6-4
Введите значение переменной x:
7
Результат: 70
```

Рис. 4.3: Запуск исполняемого файла

5 Выводы

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