

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра физико-математических и естественных наук

ОТЧЕТ

ПО ЛАБОРАТОРНОЙ РАБОТЕ № 6

дисциплина: Архитектура компьютера

Студент: Попова Алиса Владимировна

Группа: НПИ-03-25

МОСКВА

2025г.

Цель работы

Освоение арифметических инструкций языка ассемблера NASM

Задание 1

1. Создайте каталог для программ лабораторной работы № 6, перейдите в него и создайте файл lab6-1.asm.
2. Введите в файл lab6-1.asm текст программы из листинга 6.1
3. Создайте исполняемый файл и запустите его.

A terminal window with a dark background and green text. The window title is 'alisa-vladimirovna-p@alisa-vladimirovna-p-Nitro-ANV15-41: ~/work/arch-...'. The user navigates to the directory ~/work/arch-pc/lab06 and creates a file lab6-1.asm using the touch command. Then, they attempt to assemble it with nasm -f elf lab6-1.asm, which fails with an error: 'lab6-1.asm:1: error: unable to open include file `in_out.asm': No such file or directory'. They then successfully assemble it with nasm -f elf_i386 -o lab6-1 lab6-1.o. Finally, they run ./lab6-1, which produces the output 'j'.

Создан файл lab6-1.asm, введён текст из программы листинга 6.1, создан и запущен исполняемый файл. Получен ожидаемый ответ.

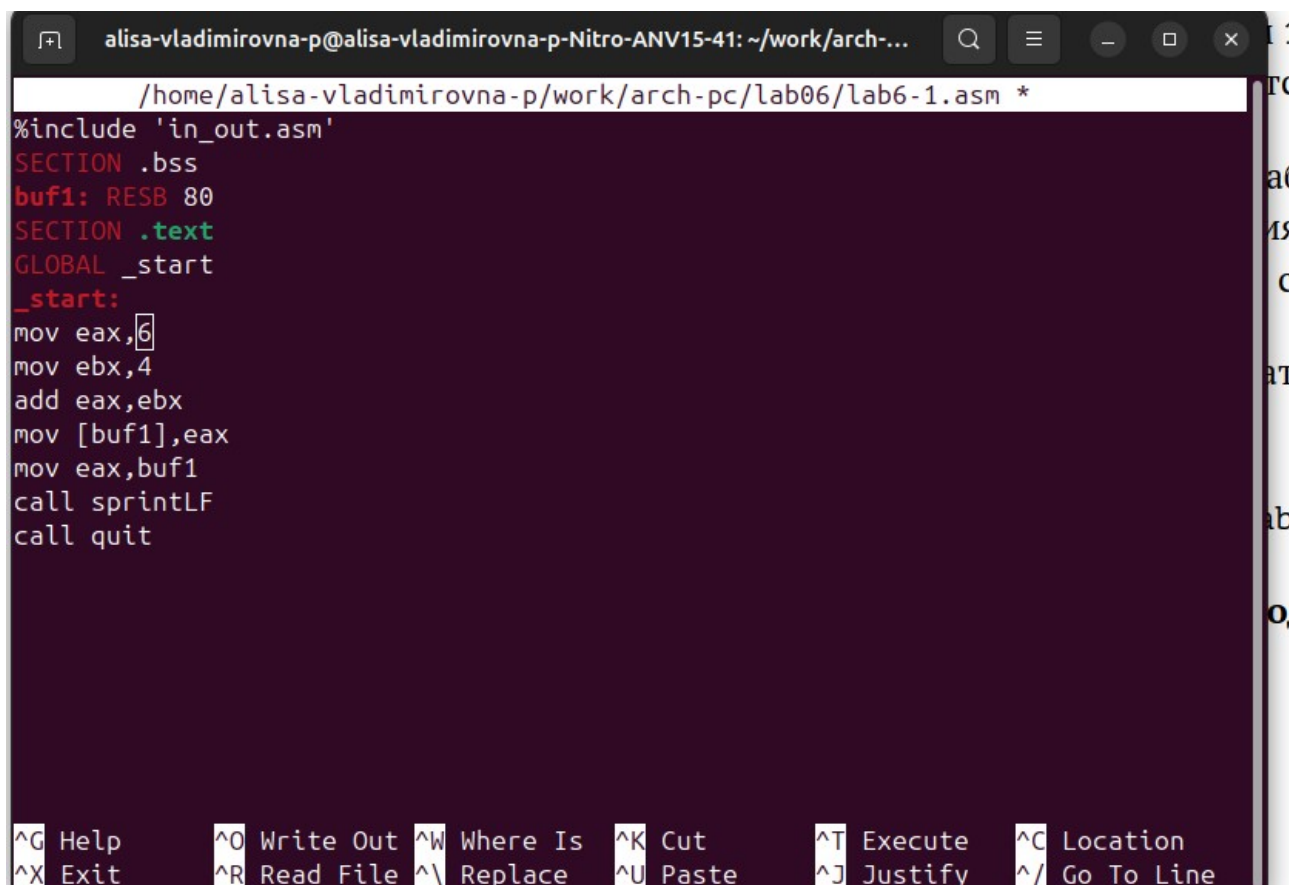
Задание 2

3. Далее изменим текст программы и вместо символов, запишем в регистры числа. Исправьте текст программы (Листинг 6.1) следующим образом: замените строки `mov eax,'6'`, `mov ebx,'4'`

на строки `mov eax,6`,

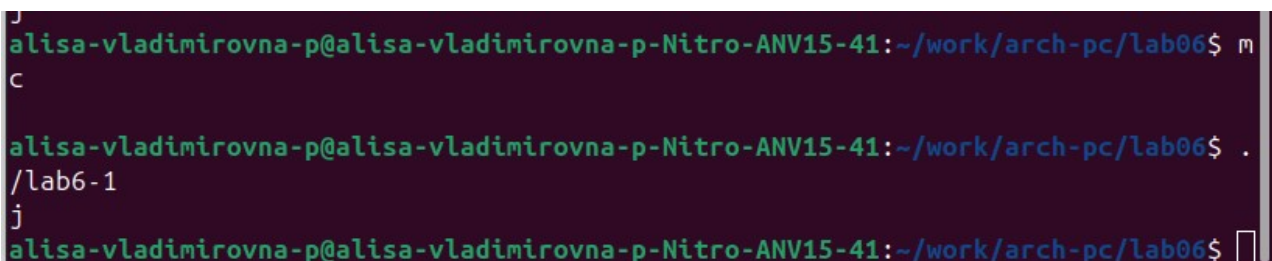
`mov ebx,4`

Создайте исполняемый файл и запустите его.



```
alisa-vladimirovna-p@alisa-vladimirovna-p-Nitro-ANV15-41: ~/work/arch-...
/home/alisa-vladimirovna-p/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

^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^/ Go To Line
```



```
alisa-vladimirovna-p@alisa-vladimirovna-p-Nitro-ANV15-41:~/work/arch-pc/lab06$ make
c

alisa-vladimirovna-p@alisa-vladimirovna-p-Nitro-ANV15-41:~/work/arch-pc/lab06$ ./lab6-1
10
alisa-vladimirovna-p@alisa-vladimirovna-p-Nitro-ANV15-41:~/work/arch-pc/lab06$
```

Задание 3

```
alisa-vladimirovna-p@alisa-vladimirovna-p-Nitro-ANV15-41: ~/work/arch-pc/lab06/lab6-2.asm
/home/alisa-vladimirovna-p/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

[ Read 9 lines ]
^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify
```

```
alisa-vladimirovna-p@alisa-vladimirovna-p-Nitro-ANV15-41:~/work/arch-pc/lab06$ ./lab6-1
j
alisa-vladimirovna-p@alisa-vladimirovna-p-Nitro-ANV15-41:~/work/arch-pc/lab06$ touch lab6-2.asm
alisa-vladimirovna-p@alisa-vladimirovna-p-Nitro-ANV15-41:~/work/arch-pc/lab06$ mc
alisa-vladimirovna-p@alisa-vladimirovna-p-Nitro-ANV15-41:~/work/arch-pc/lab06$ nasm -f elf lab6-2.asm
alisa-vladimirovna-p@alisa-vladimirovna-p-Nitro-ANV15-41:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-2 lab6-2.o
ld: unrecognized emulation mode: elf_i386
Supported emulations: elf_x86_64 elf32_x86_64 elf_i386 elf_iamcu i386pep i386pe
alisa-vladimirovna-p@alisa-vladimirovna-p-Nitro-ANV15-41:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-2 lab6-2.o
alisa-vladimirovna-p@alisa-vladimirovna-p-Nitro-ANV15-41:~/work/arch-pc/lab06$ ./lab6-2
106
alisa-vladimirovna-p@alisa-vladimirovna-p-Nitro-ANV15-41:~/work/arch-pc/lab06$
```

Задание 4

```
alisa-vladimirovna-p@alisa-vladimirovna-p-Nitro-ANV15-41: ~/work/ar...
/home/alisa-vladimirovna-p/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: '
    mov eax,edi ; вызов подпрограммы печати значения

^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify
```

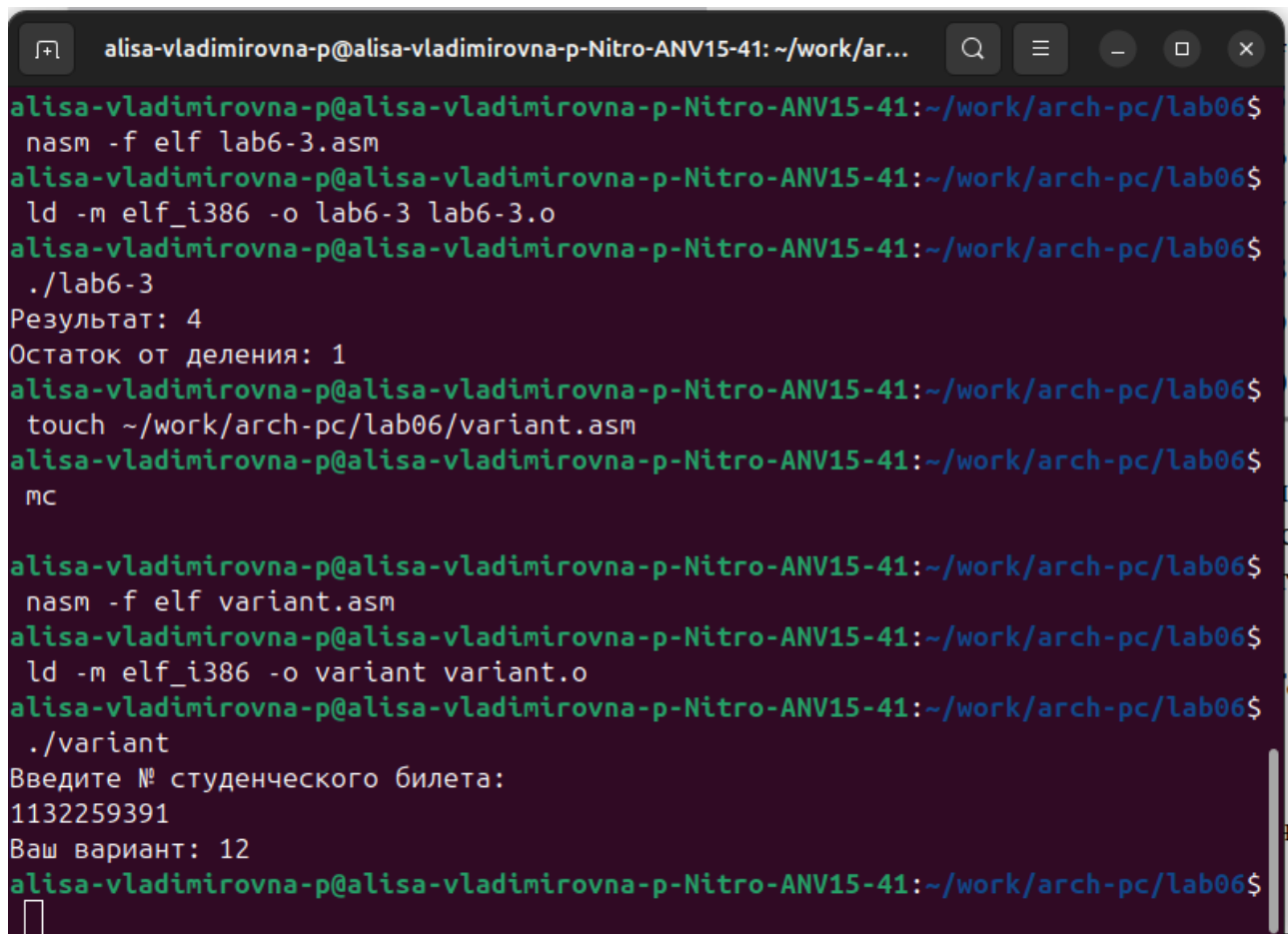
```
alisa-vladimirovna-p@alisa-vladimirovna-p-Nitro-ANV15-41: ~/work/ar...
alisa-vladimirovna-p@alisa-vladimirovna-p-Nitro-ANV15-41:~/work/arch-pc/lab06$
mc
alisa-vladimirovna-p@alisa-vladimirovna-p-Nitro-ANV15-41:~/work/arch-pc/lab06$
    nasm -f elf lab6-3.asm
lab6-3.asm:8: error: label or instruction expected at start of line
alisa-vladimirovna-p@alisa-vladimirovna-p-Nitro-ANV15-41:~/work/arch-pc/lab06$
    nasm -f elf lab6-3.asm
lab6-3.asm:8: error: label or instruction expected at start of line
alisa-vladimirovna-p@alisa-vladimirovna-p-Nitro-ANV15-41:~/work/arch-pc/lab06$
    nasm -f elf lab6-3.asm
lab6-3.asm:8: error: label or instruction expected at start of line
alisa-vladimirovna-p@alisa-vladimirovna-p-Nitro-ANV15-41:~/work/arch-pc/lab06$
    nano lab6-3.asm
alisa-vladimirovna-p@alisa-vladimirovna-p-Nitro-ANV15-41:~/work/arch-pc/lab06$
    nasm -f elf lab6-3.asm
alisa-vladimirovna-p@alisa-vladimirovna-p-Nitro-ANV15-41:~/work/arch-pc/lab06$
    ld -m elf_i386 -o lab6-3 lab6-3.o
alisa-vladimirovna-p@alisa-vladimirovna-p-Nitro-ANV15-41:~/work/arch-pc/lab06$
    ./lab6-3
Результат: 4
Остаток от деления: 1
alisa-vladimirovna-p@alisa-vladimirovna-p-Nitro-ANV15-41:~/work/arch-pc/lab06$
█
```

Задание 5

В качестве другого примера рассмотрим программу вычисления варианта задания по

номеру студенческого билета, работающую по следующему алгоритму:

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



```
alisa-vladimirovna-p@alisa-vladimirovna-p-Nitro-ANV15-41: ~/work/ar...
alisa-vladimirovna-p@alisa-vladimirovna-p-Nitro-ANV15-41:~/work/arch-pc/lab06$
nasm -f elf lab6-3.asm
alisa-vladimirovna-p@alisa-vladimirovna-p-Nitro-ANV15-41:~/work/arch-pc/lab06$
ld -m elf_i386 -o lab6-3 lab6-3.o
alisa-vladimirovna-p@alisa-vladimirovna-p-Nitro-ANV15-41:~/work/arch-pc/lab06$
./lab6-3
Результат: 4
Остаток от деления: 1
alisa-vladimirovna-p@alisa-vladimirovna-p-Nitro-ANV15-41:~/work/arch-pc/lab06$
touch ~/work/arch-pc/lab06/variant.asm
alisa-vladimirovna-p@alisa-vladimirovna-p-Nitro-ANV15-41:~/work/arch-pc/lab06$
mc
alisa-vladimirovna-p@alisa-vladimirovna-p-Nitro-ANV15-41:~/work/arch-pc/lab06$
nasm -f elf variant.asm
alisa-vladimirovna-p@alisa-vladimirovna-p-Nitro-ANV15-41:~/work/arch-pc/lab06$
ld -m elf_i386 -o variant variant.o
alisa-vladimirovna-p@alisa-vladimirovna-p-Nitro-ANV15-41:~/work/arch-pc/lab06$
./variant
Введите № студенческого билета:
1132259391
Ваш вариант: 12
alisa-vladimirovna-p@alisa-vladimirovna-p-Nitro-ANV15-41:~/work/arch-pc/lab06$
```

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

`mov eax, rem` - `rem` содержит строку 'Ваш вариант: '

`call sprintLF` - вывод строки и перевода строки

2. Для чего используются следующие инструкции?

```
mov ecx, x
```

```
mov edx, 80  
call sread
```

- **mov ecx, x** — помещает в регистр ecx адрес буфера x (куда будут записаны введенные данные).
- **mov edx, 80** — помещает в регистр edx максимальное количество байт для чтения (80).
- **call sread** — вызывает функцию чтения строки с клавиатуры (стандартный ввод). Введенная строка сохраняется по адресу в ecx.

3. Для чего используется инструкция “call atoi”?

Инструкция `call atoi` вызывает функцию преобразования ASCII-строки в целое число.

Перед вызовом в `eax` должен быть адрес строки. После выполнения в `eax` будет целочисленный результат.

В программе это нужно, чтобы преобразовать введенный номер студенческого билета (строку) в число для дальнейших вычислений.

4. Какие строки листинга 6.4 отвечают за вычисления варианта?

```
mov eax,x
```

```
call atoi
```

```
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 `div ebx` :

- **Частное** записывается в регистр `eax`
- **Остаток** записывается в регистр `edx`

6. Для чего используется инструкция “`inc edx`”?

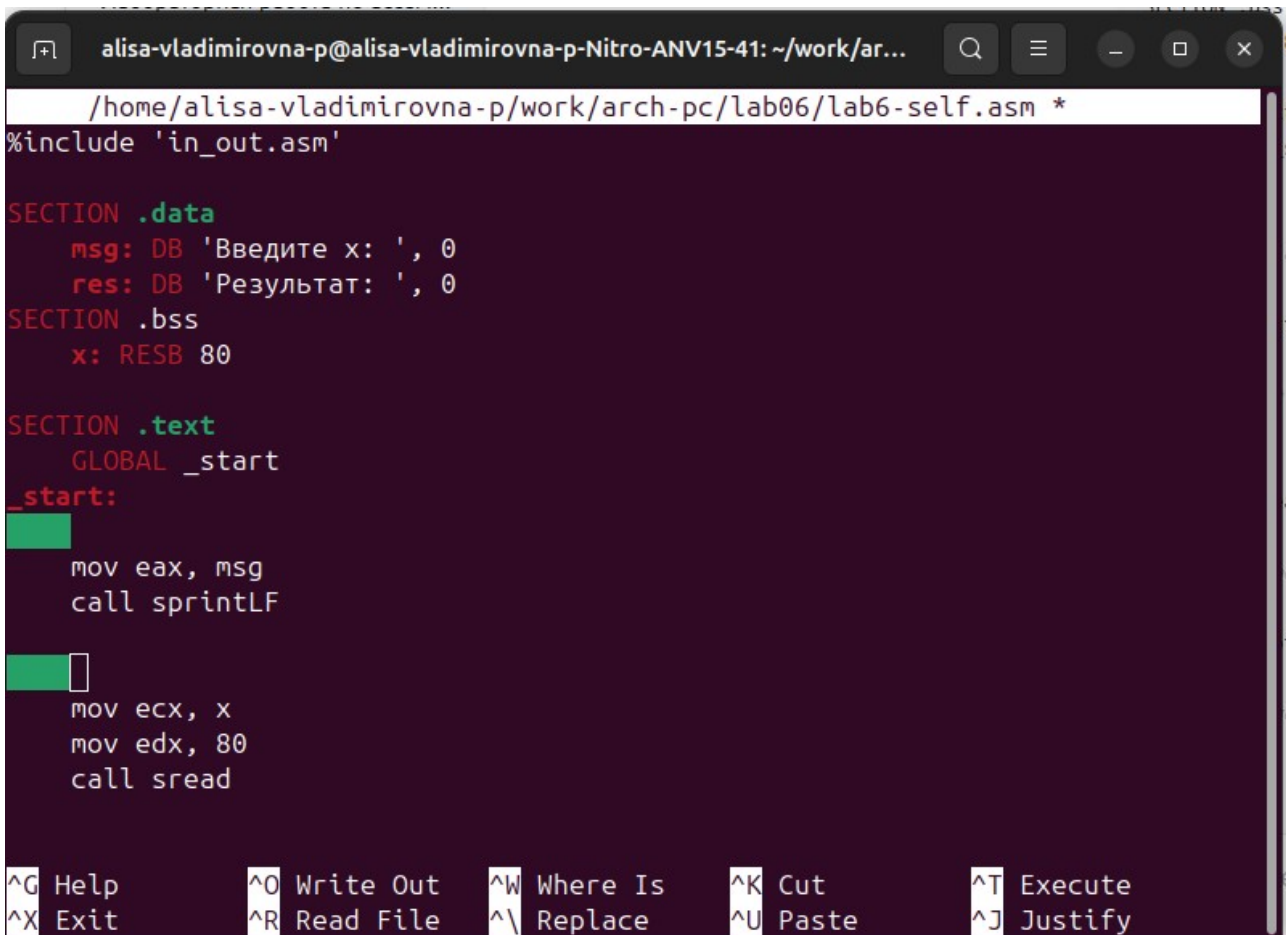
Инструкция `inc edx` увеличивает значение в регистре `edx` на 1.

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

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

Задание для самостоятельной работы

1. Написать программу вычисления выражения $y = f(x)$. Программа должна выводить выражение для вычисления, выводить запрос на ввод значения x , вычислять заданное выражение в зависимости от введенного x , выводить результат вычислений. Вид функции $f(x)$ выбрать из таблицы 6.3 вариантов заданий в соответствии с номером полученным при выполнении лабораторной работы. Создайте исполняемый файл и проверьте его работу для значений x_1 и x_2 из 6.3.



```
alisa-vladimirovna-p@alisa-vladimirovna-p-Nitro-ANV15-41: ~/work/ar...
/home/alisa-vladimirovna-p/work/arch-pc/lab06/lab6-self.asm *
#include 'in_out.asm'

SECTION .data
    msg: DB 'Введите x: ', 0
    res: 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

^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify
```

Написана программа удовлетворяющая условию задачи 12 $(8x - 6)/2$:

```
%include 'in_out.asm'

SECTION .data
    msg: DB 'Введите x: ', 0
    res: DB 'Результат: ', 0

SECTION .bss
    x: RESB 80

SECTION .text
    GLOBAL _start

_start:
    ; Вывод сообщения "Введите x: "
    mov eax, msg
    call sprintf
```

```

; Чтение введённого значения (строка)
mov ecx, x
mov edx, 80
call sread

; Преобразование строки в число (x)
mov eax, x
call atoi          ; теперь eax = x (число)

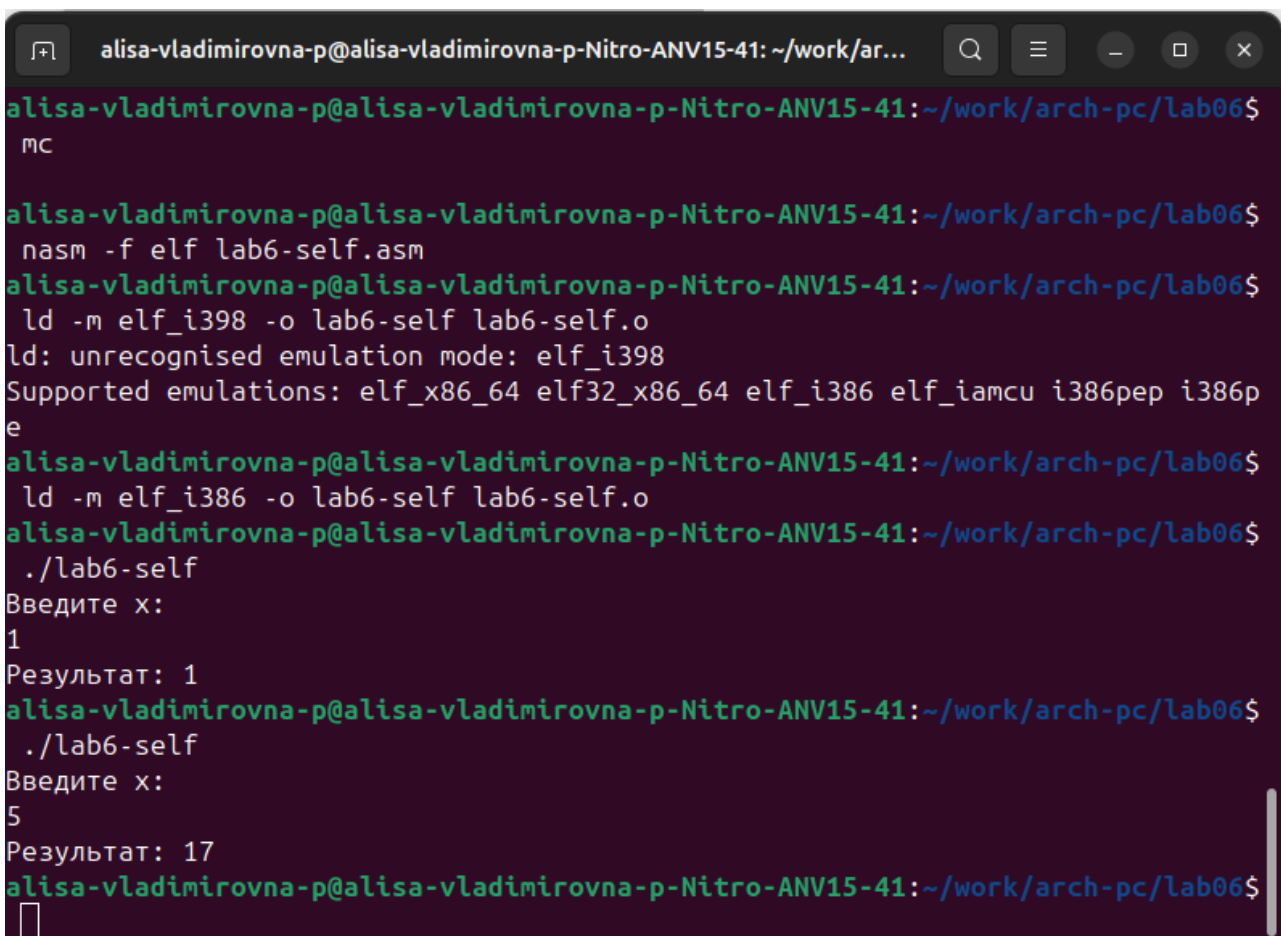
; Вычисление  $f(x) = (8 \cdot x - 6) / 2$ 
mov ebx, 8
mul ebx            ; eax = 8 * x
sub eax, 6         ; eax = 8 * x - 6
mov ebx, 2
div ebx            ; eax = (8 * x - 6) / 2 (частное)

; Вывод результата
mov ecx, eax       ; сохраняем результат в ecx
mov eax, res       ; "Результат: "
call sprint
mov eax, ecx       ; восстанавливаем результат
call iprintLF      ; вывод числа

; Завершение программы
call quit

```

Запуск программы (вариант 12)



```

alisa-vladimirovna-p@alisa-vladimirovna-p-Nitro-ANV15-41: ~/work/arch-pc/lab06$
mc

alisa-vladimirovna-p@alisa-vladimirovna-p-Nitro-ANV15-41:~/work/arch-pc/lab06$
nasm -f elf lab6-self.asm
alisa-vladimirovna-p@alisa-vladimirovna-p-Nitro-ANV15-41:~/work/arch-pc/lab06$
ld -m elf_i398 -o lab6-self lab6-self.o
ld: unrecognized emulation mode: elf_i398
Supported emulations: elf_x86_64 elf32_x86_64 elf_i386 elf_iamcu i386pep i386p
e
alisa-vladimirovna-p@alisa-vladimirovna-p-Nitro-ANV15-41:~/work/arch-pc/lab06$
ld -m elf_i386 -o lab6-self lab6-self.o
alisa-vladimirovna-p@alisa-vladimirovna-p-Nitro-ANV15-41:~/work/arch-pc/lab06$
./lab6-self
Введите x:
1
Результат: 1
alisa-vladimirovna-p@alisa-vladimirovna-p-Nitro-ANV15-41:~/work/arch-pc/lab06$
./lab6-self
Введите x:
5
Результат: 17
alisa-vladimirovna-p@alisa-vladimirovna-p-Nitro-ANV15-41:~/work/arch-pc/lab06$

```

Вывод

В ходе выполнения лабораторной работы №6 были успешно освоены арифметические инструкции языка ассемблера NASM и принципы работы с символьными и числовыми данными.

Список литературы

1. GDB: The GNU Project Debugger. — URL: <https://www.gnu.org/software/gdb/>.
2. GNU Bash Manual. — 2016. — URL: <https://www.gnu.org/software/bash/manual/>.
3. Midnight Commander Development Center. — 2021. — URL: <https://midnight-commander.org/>.
4. NASM Assembly Language Tutorials. — 2021. — URL: <https://asmtutor.com/>.
5. Newham C. Learning the bash Shell: Unix Shell Programming. — O'Reilly Media, 2005. — 354 с. — (In a Nutshell). — ISBN 0596009658. — URL: <http://www.amazon.com/Learning-bash-Shell-Programming-Nutshell/dp/0596009658>.
6. Robbins A. Bash Pocket Reference. — O'Reilly Media, 2016. — 156 с. — ISBN 978-1491941591.
7. The NASM documentation. — 2021. — URL: <https://www.nasm.us/docs.php>.
8. Zarrelli G. Mastering Bash. — Packt Publishing, 2017. — 502 с. — ISBN 9781784396879.
9. Колдаев В. Д., Лупин С. А. Архитектура ЭВМ. — М. : Форум, 2018.
10. Куляс О. Л., Никитин К. А. Курс программирования на ASSEMBLER. — М. : Солон-Пресс, 2017.
11. Новожилов О. П. Архитектура ЭВМ и систем. — М. : Юрайт, 2016.
12. Расширенный ассемблер: NASM. — 2021. — URL: <https://www.opennet.ru/docs/RUS/nasm/>.
13. Робачевский А., Немнюгин С., Стесик О. Операционная система UNIX. — 2-е изд. — БХВ-Петербург, 2010. — 656 с. — ISBN 978-5-94157-538-1.
14. Столяров А. Программирование на языке ассемблера NASM для ОС Unix. — 2-е изд. — М. : МАКС Пресс, 2011. — URL: http://www.stolyarov.info/books/asm_unix.
15. Таненбаум Э. Архитектура компьютера. — 6-е изд. — СПб. : Питер, 2013. — 874 с. — (Классика Computer Science).
16. Таненбаум Э., Бос Х. Современные операционные системы. — 4-е изд. — СПб. : Питер, 2015. — 1120 с. — (Классика Computer Science)