

Лабораторная работа №6

Арифметические операции в NASM

Баранов Никита Дмитриевич

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
4	Ответы на вопросы	14
5	Самостоятельная работа	15
6	Выводы	17

Список иллюстраций

3.1	Создаем файл и каталог для него	7
3.2	Вписываем программу в файл	7
3.3	Создаем исполняемый файл и проверяем работу	8
3.4	Уберем кавычки из текста программы и проверим работу	8
3.5	Создаем файл lab6-2.asm	8
3.6	Вводим программу	9
3.7	Создаем исполняемый файл и запускаем его	9
3.8	Создаем исполняемый файл для измененного файла(изменяем строки на числа) и запускаем его	9
3.9	Меняем <code>iprintLF</code> на <code>iprint</code>	10
3.10	Создаем исполняемый файл и запускаем его	10
3.11	Создаем файл и вводим туда программу	11
3.12	Создаем исполняемый файл и смотрим результат	11
3.13	Изменяем программу	12
3.14	Создаем исполняемый файл и смотрим результат	12
3.15	Создаем файл и вводим программу	13
3.16	Создаем исполняемый файл и смотрим результат программы . . .	13
5.1	Создаем новый файл и вводим программу чтобы решалось выражение $f(x) = (x+2)^2$	16
5.2	Создаем исполняемый файл и проверяем работу программы . . .	16

Список таблиц

1 Цель работы

Освоение арифметических инструкций языка ассемблера NASM.

2 Задание

Написать программы для решения выражений.

3 Выполнение лабораторной работы

Создайте каталог для программ лабораторной работы № 6, перейдите в него и создайте файл lab6-1.asm . Введите в файл lab6-1.asm текст программы из листинга 6.1. Далее выводим результат. Создайте исполняемый файл и запустите его. Далее изменим текст программы и вместо символов, запишем в регистры числа(рис. fig. 3.1)(рис. fig. 3.2)(рис. fig. 3.3)(рис. fig. 3.4)

```
baranov@fedora:~$ mkdir ~/work/arch-pc/lab06
baranov@fedora:~$ cd ~/work/arch-pc/lab06
baranov@fedora:~/work/arch-pc/lab06$ touch lab6-1.asm
baranov@fedora:~/work/arch-pc/lab06$ mc
```

Рис. 3.1: Создаем файл и каталог для него



```
GNU nano 7.2 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 sprintLF
call quit
```

[Прочитано 14 строк]

^G Справка ^O Записать ^W Поиск ^K Вырезать ^T Выполнить ^C Позиция
^X Выход ^R ЧитФайл ^\ Замена ^U Вставить ^J Выровнять ^/ К строке

Рис. 3.2: Вписываем программу в файл

```

baranov@fedora:~/work/arch-pc/lab06$ nano lab6-1.asm
baranov@fedora:~/work/arch-pc/lab06$ nasm -f elf lab6-1.asm
baranov@fedora:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-1 lab6-1.o
baranov@fedora:~/work/arch-pc/lab06$ ./lab6-1
j

```

Рис. 3.3: Создаем исполняемый файл и проверяем работу

```

baranov@fedora:~/work/arch-pc/lab06$ nasm -f elf lab6-1.asm
baranov@fedora:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-1 lab6-1.o
baranov@fedora:~/work/arch-pc/lab06$ ./lab6-1

```

Рис. 3.4: Уберем кавычки из текста программы и проверим работу

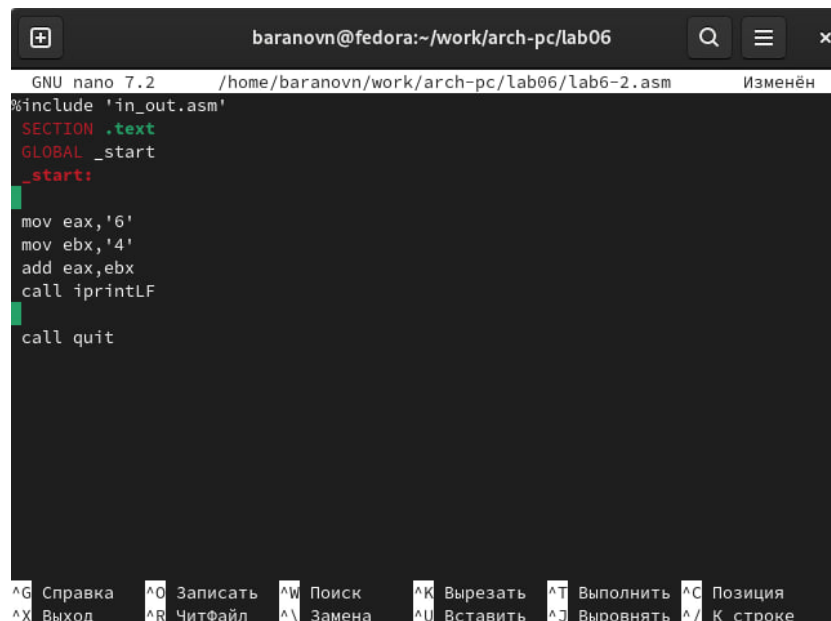
Создайте файл lab6-2.asm в каталоге ~/work/arch-pc/lab06 и введите в него текст программы из листинга 6.2. Создайте исполняемый файл и запустите его. Аналогично предыдущему примеру изменим символы на числа. Создайте исполняемый файл и запустите его. Какой результат будет получен при исполнении программы? Замените функцию `iprintLF` на `iprint`. Создайте исполняемый файл и запустите его. Чем отличается вывод функций `iprintLF` и `iprint`? (рис. fig. 3.5)(рис. fig. 3.6)(рис. fig. 3.7)(рис. fig. 3.8)(рис. fig. 3.9)(рис. fig. 3.10)

```

baranov@fedora:~/work/arch-pc/lab06$ touch lab6-2.asm
baranov@fedora:~/work/arch-pc/lab06$ touch lab6-2.asm

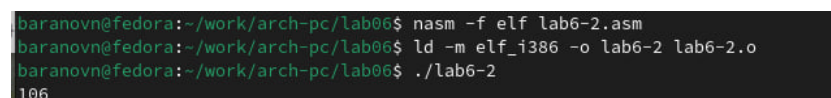
```

Рис. 3.5: Создаем файл lab6-2.asm



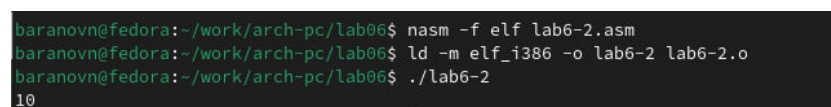
```
baranovn@fedora:~/work/arch-pc/lab06
GNU nano 7.2 /home/baranovn/work/arch-pc/lab06/lab6-2.asm
%include 'in_out.asm'
SECTION .text
GLOBAL _start
_start:
mov eax,'6'
mov ebx,'4'
add eax,ebx
call iprintLF
call quit
```

Рис. 3.6: Вводим программу



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

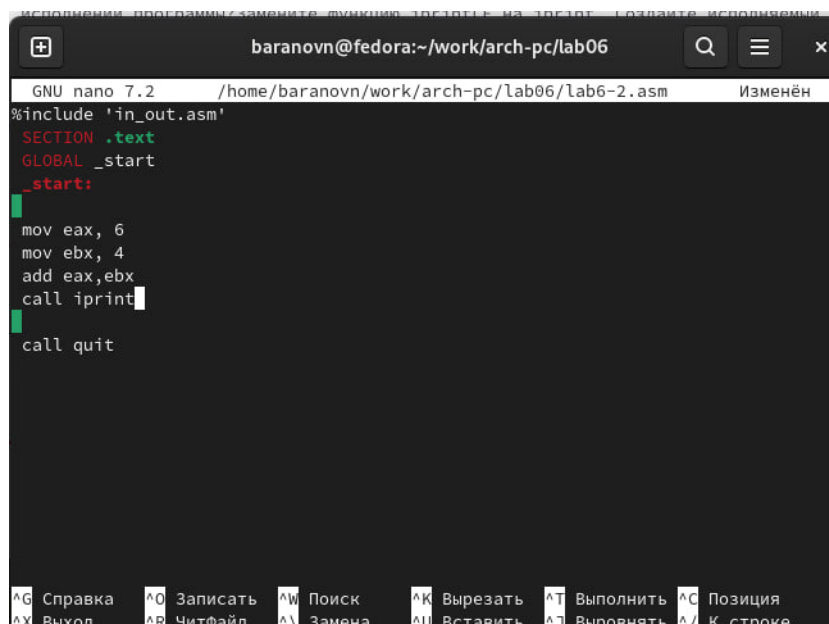
Рис. 3.7: Создаем исполняемый файл и запускаем его



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


Рис. 3.8: Создаем исполняемый файл для измененного файла(изменяем строки на числа) и запускаем его

В результате программы мы наконец получили сумму 6 и 4 - 10.



```
baranovn@fedora:~/work/arch-pc/lab06
GNU nano 7.2 /home/baranovn/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 iprint
call quit
```

Рис. 3.9: Меняем `iprintLF` на `iprint`

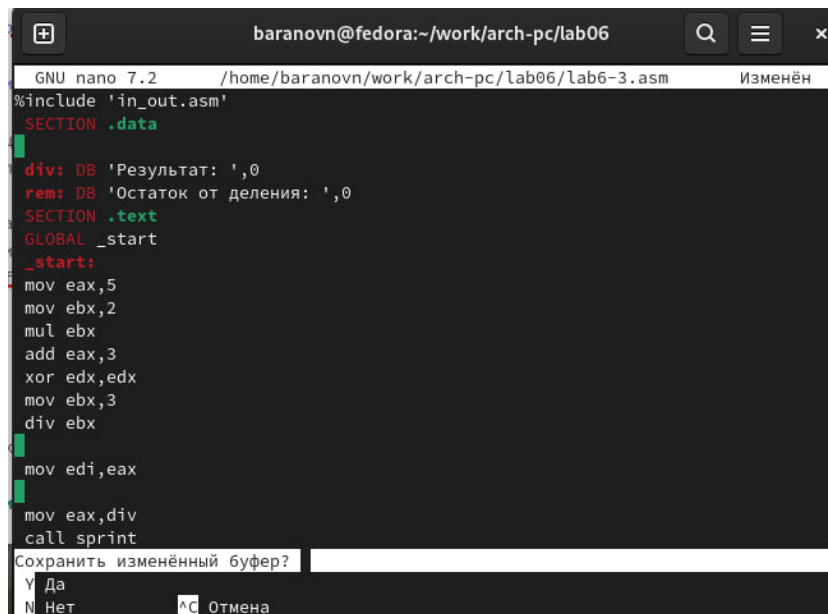


```
baranovn@fedora:~/work/arch-pc/lab06$ nasm -f elf lab6-2.asm
baranovn@fedora:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-2 lab6-2.o
baranovn@fedora:~/work/arch-pc/lab06$ ./lab6-2
10baranovn@fedora:~/work/arch-pc/lab06$
```

Рис. 3.10: Создаем исполняемый файл и запускаем его

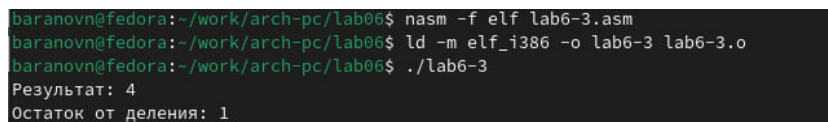
Вывод функций различается тем, что `iprintLF` в отличие от `iprint` не переносит на другую строку после вывода.

В качестве примера выполнения арифметических операций в NASM приведем программу вычисления арифметического выражения $\square(\square) = (5 \square 2 + 3)/3$. Создайте файл `lab6-3.asm` в каталоге `~/work/arch-pc/lab06`. Внимательно изучите текст программы из листинга 6.3 и введите в `lab6-3.asm`. Создайте исполняемый файл и запустите его. Измените текст программы для вычисления выражения $\square(\square) = (4 \square 6 + 2)/5$. Создайте исполняемый файл и проверьте его работу (рис. fig. 3.11) (рис. fig. 3.12) (рис. fig. 3.13) (рис. fig. 3.14)



```
baranovn@fedora:~/work/arch-pc/lab06
GNU nano 7.2 /home/baranovn/work/arch-pc/lab06/lab6-3.asm
%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
Сохранить изменённый буфер?
Y Да
N Нет ^C Отмена
```

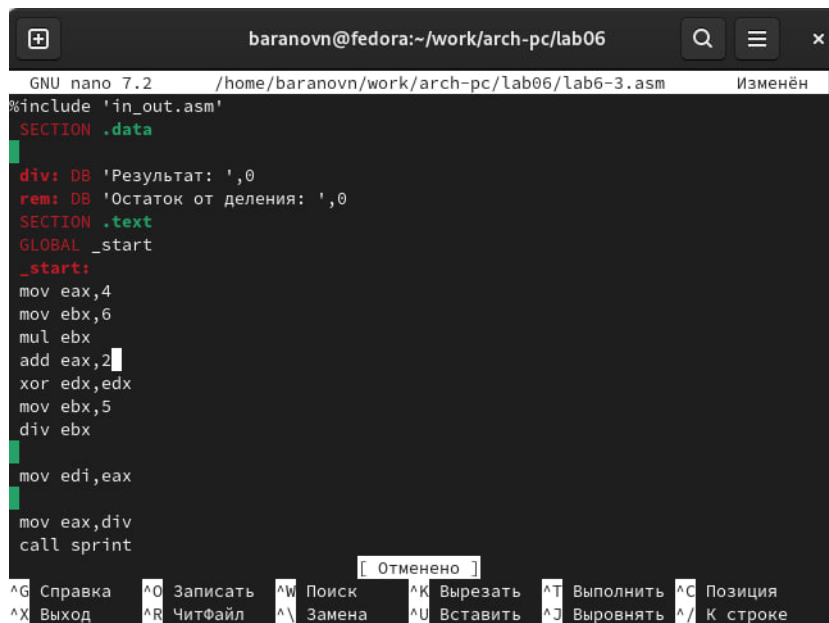
Рис. 3.11: Создаем файл и вводим туда программу



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

Рис. 3.12: Создаем исполняемый файл и смотрим результат

Наш результат совпал с тем, что должен был получиться.

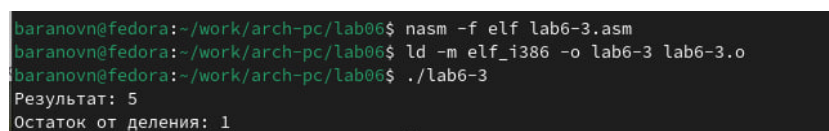


```
GNU nano 7.2 /home/baranovn/work/arch-pc/lab06/lab6-3.asm
%include 'in_out.asm'
SECTION .data
div: DB 'Результат: ',0
rem: DB 'Остаток от деления: ',0
SECTION .text
GLOBAL _start
_start:
mov eax,4
mov ebx,6
mul ebx
add eax,2
xor edx,edx
mov ebx,5
div ebx
mov edi,eax
mov eax,div
call sprint
```

Отменено

^G Справка ^O Записать ^W Поиск ^K Вырезать ^T Выполнить ^C Позиция
^X Выход ^R ЧитФайл ^\ Замена ^U Вставить ^J Выводить ^/_ К строке

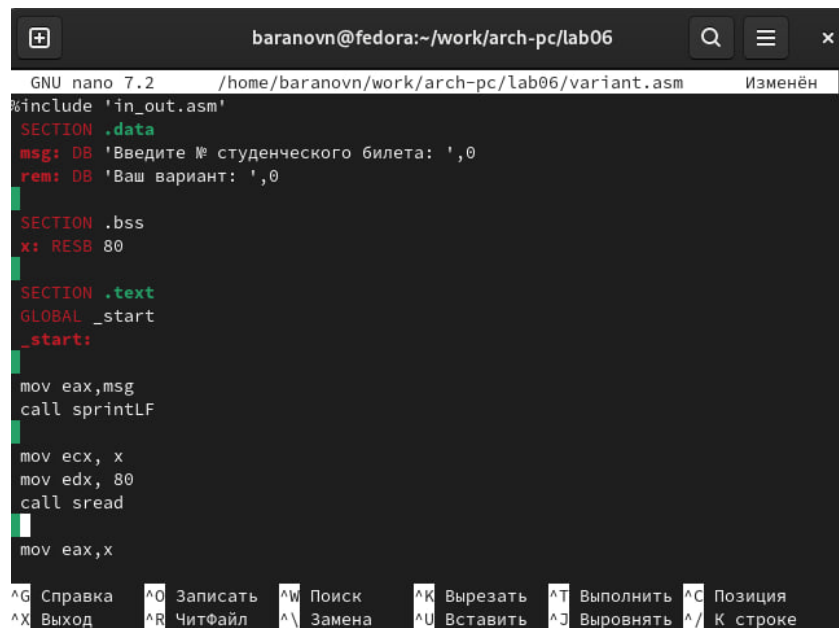
Рис. 3.13: Изменяем программу



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

Рис. 3.14: Создаем исполняемый файл и смотрим результат

Создайте файл `variant.asm` в каталоге `~/work/arch-pc/lab06`. Внимательно изучите текст программы из листинга 6.4 и введите в файл `variant.asm`. Создайте исполняемый файл и запустите его. Проверьте результат работы программы вычислив номер варианта аналитически.(рис. fig. 3.15)(рис. fig. 3.16)



```
GNU nano 7.2 /home/baranovn/work/arch-pc/lab06/variant.asm
%include 'in_out.asm'
SECTION .data
msg: DB 'Введите № студенческого билета: ',0
rem: DB 'Ваш вариант: ',0

SECTION .bss
x: RESB 80

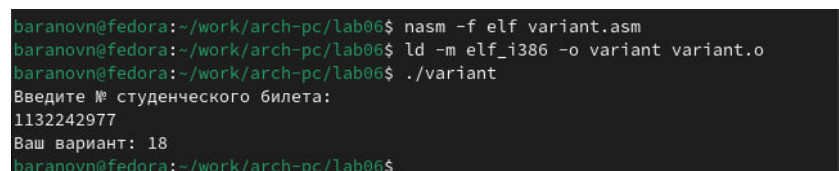
SECTION .text
GLOBAL _start
_start:

mov eax,msg
call sprintLF

mov ecx, x
mov edx, 80
call sread

mov eax,x
```

Рис. 3.15: Создаем файл и вводим программу



```
baranovn@fedora:~/work/arch-pc/lab06$ nasm -f elf variant.asm
baranovn@fedora:~/work/arch-pc/lab06$ ld -m elf_i386 -o variant variant.o
baranovn@fedora:~/work/arch-pc/lab06$ ./variant
Введите № студенческого билета:
1132242977
Ваш вариант: 18
baranovn@fedora:~/work/arch-pc/lab06$
```

Рис. 3.16: Создаем исполняемый файл и смотрим результат программы

4 Ответы на вопросы

1. Строки `"mov eax,rem"`, `"call sprint"` отвечают за вывод сообщения "Ваш вариант" на экран
2. Эти инструкции позволяют программе считать строчки которые пользователь вводит на экран
3. Инструкция `"call atoi"` преобразует строку в целое число
4. Строка `"xor edx,edx"` обнуляет регистр `edx` перед выполнением деления. Строка `"mov ebx,20"` загружает значение 20 в регистр `ebx`. Строка `"div ebx"` выполняет деление регистра `eax` на значение регистра `ebx` с сохранением частного в регистре `eax` и остатка в регистре `edx`.
5. Остаток от деления записывает в регистр `edx`
6. Функция используется для увеличения значения регистра `edx` на 1
7. Строки `"mov eax, edx"`, `"call iprintLF"`. Первая передает значение остатка от деления в регистр `eax`. Вторая выводит значения на экран.

5 Самостоятельная работа

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(рис. fig. 5.1)(рис. fig. 5.2)



```
baranovn@fedora:~/work/arch-pc/lab06
GNU nano 7.2 /home/baranovn/work/arch-pc/lab06/homework.asm
#include 'in_out.asm'

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

SECTION .bss
rez: RESB 80
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

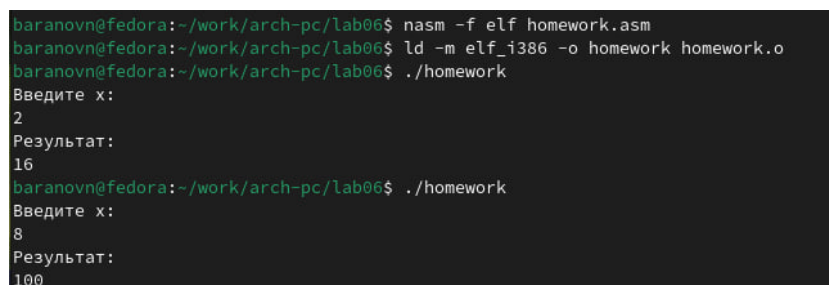
    add eax, 2

    mov ebx, eax
    mul ebx

    mov [rez], eax

    mov eax, rem
    call sprintf
```

Рис. 5.1: Создаем новый файл и вводим программу чтобы решалось выражение $f(x) = (x+2)^2$



```
baranovn@fedora:~/work/arch-pc/lab06$ nasm -f elf homework.asm
baranovn@fedora:~/work/arch-pc/lab06$ ld -m elf_i386 -o homework homework.o
baranovn@fedora:~/work/arch-pc/lab06$ ./homework
Введите x:
2
Результат:
16
baranovn@fedora:~/work/arch-pc/lab06$ ./homework
Введите x:
8
Результат:
100
```

Рис. 5.2: Создаем исполняемый файл и проверяем работу программы

6 Выводы

Мы приобрели навыки создания исполнительных файлов для решения выражений и освоили арифметические инструкции в NASM.