

Отчёт по лабораторной работе №6

Дисциплина: Архитектура Компьютеров

Мургия Марк Максимович

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	8
5	Выводы	12
	Список литературы	13

Список иллюстраций

4.1	lab6-1.asm	8
4.2	lab6-2.asm	9
4.3	Результаты lab6-2.asm	9
4.4	lab6-3.asm	10
4.5	Результаты lab6-3.asm	10
4.6	lab6-4.asm работает	10

Список таблиц

3.1 Арифметические операции в NASM	7
--	---

1 Цель работы

Освоение арифметических инструкций языка ассемблера NASM.

2 Задание

1. Понять работу арифметических операций
2. Использовать из на практике, создавая ассемблерские программы

3 Теоретическое введение

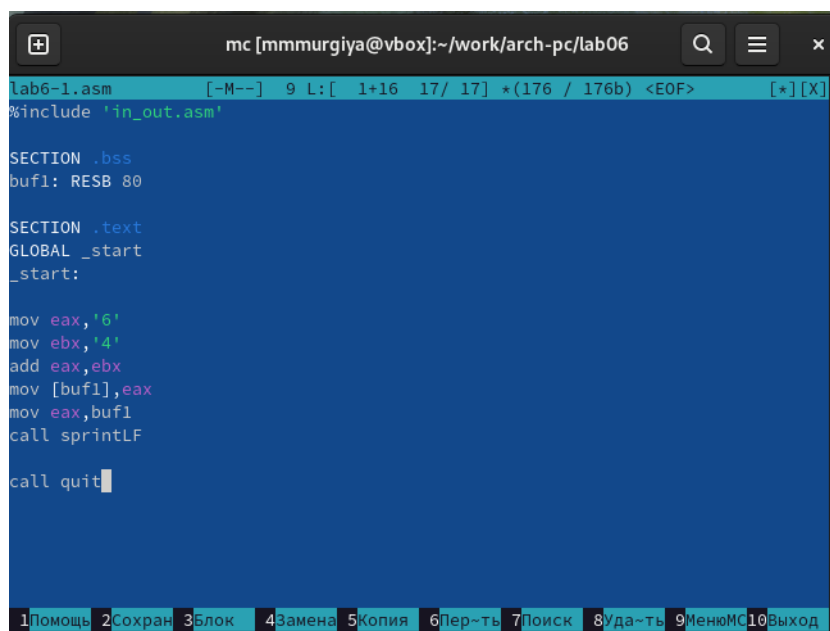
В табл. 3.1 приведены все операции, которыми пользовался я в данной лабораторной работе.

Таблица 3.1: Арифметические операции в NASM

Имя операции	Описание операции
add	Целочисленное сложение
sub	Целочисленное вычитание
inc	Прибавление единицы
dec	Вычитание единицы
neg	Изменение знака
mul	Беззнаковое умножение
imul	Знаковое умножение
div	Беззнаковое деление
idiv	Знаковое деление

4 Выполнение лабораторной работы

Как говорится в инструкции по лабораторной работе №6, нужно создать несколько файлов, показывающих виды вывода результатов. В первом варианте выводится символ ASCII, так как мы сложили индексы цифр 6 и 4.



```
lab6-1.asm [-M--] 9 L: [ 1+16 17/ 17] *(176 / 176b) <EOF> [*][X]
#include 'in_out.asm'

SECTION .bss
buf1: RESB 80

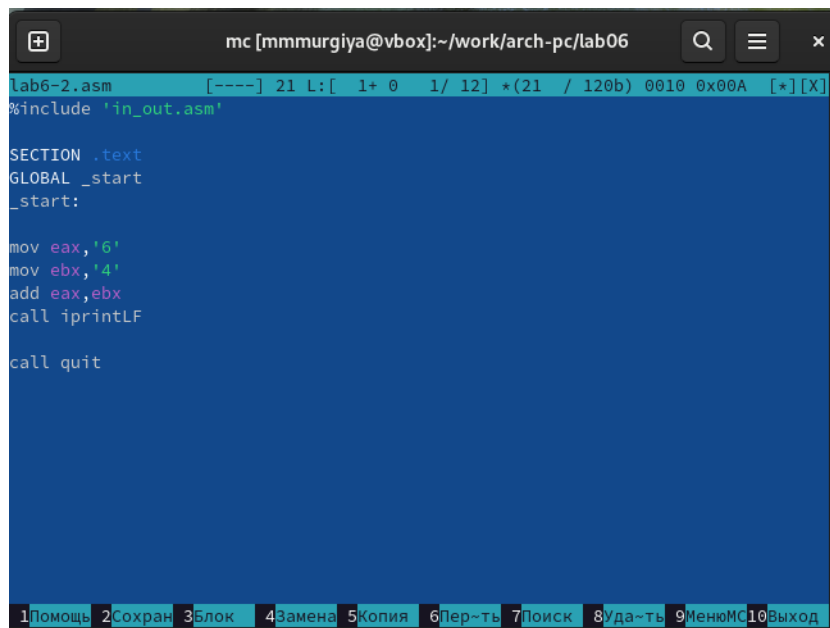
SECTION .text
GLOBAL _start
_start:

mov eax,'6'
mov ebx,'4'
add eax,ebx
mov [buf1],eax
mov eax,buf1
call sprintLF

call quit
```

Рис. 4.1: lab6-1.asm

Во втором мы получаем сам индекс. Если поменять '6' и '4' на 6 и 4, то мы получим сумму этих чисел.



```
lab6-2.asm [----] 21 L: [ 1+ 0 1/ 12] *(21 / 120b) 0010 0x00A [*] [X]
#include 'in_out.asm'

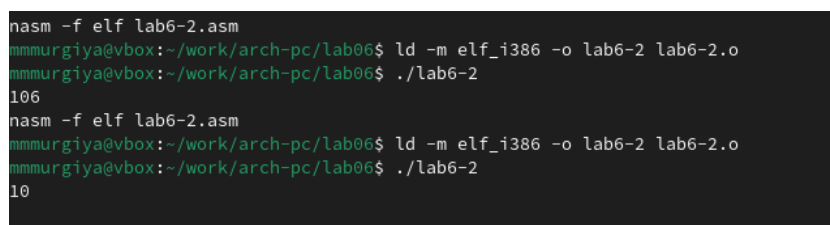
SECTION .text
GLOBAL _start
_start:

mov eax,'6'
mov ebx,'4'
add eax,ebx
call iprintLF

call quit
```

1Помощь 2Сохран 3Блок 4Замена 5Копия 6Пер-ть 7Поиск 8Уда-ть 9Меню 10Выход

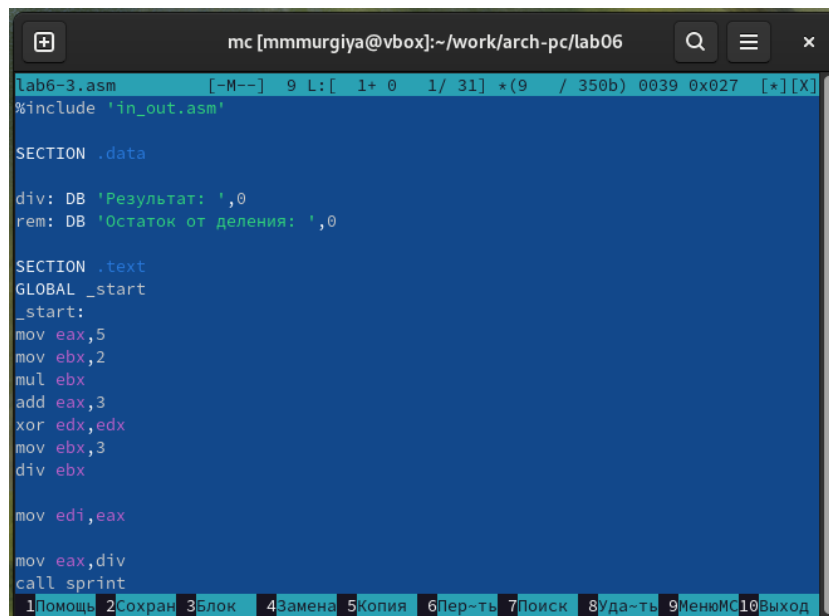
Рис. 4.2: lab6-2.asm



```
nasm -f elf lab6-2.asm
mmmurgiya@vbox:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-2 lab6-2.o
mmmurgiya@vbox:~/work/arch-pc/lab06$ ./lab6-2
106
nasm -f elf lab6-2.asm
mmmurgiya@vbox:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-2 lab6-2.o
mmmurgiya@vbox:~/work/arch-pc/lab06$ ./lab6-2
10
```

Рис. 4.3: Результаты lab6-2.asm

На фотографий ниже показывают арифметические выражения.



```
lab6-3.asm [-M--] 9 L: [ 1+ 0 1/ 31] *(9 / 350b) 0039 0x027 [*][X]
#include 'in_out.asm'

SECTION .data
div: DB 'Результат: ',0
rem: DB 'Остаток от деления: ',0

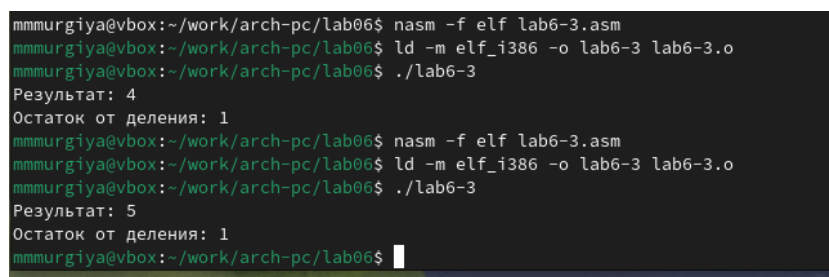
SECTION .text
GLOBAL _start
_start:
mov eax,5
mov ebx,2
mul ebx
add eax,3
xor edx,edx
mov ebx,3
div ebx

mov edi,eax

mov eax,div
call sprint

1Помощь 2Сохран 3Блок 4Замена 5Копия 6Пер-ть 7Поиск 8Уда-ть 9МенюМС10Выход
```

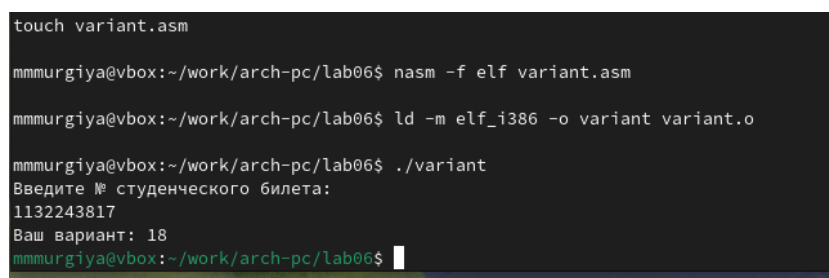
Рис. 4.4: lab6-3.asm



```
mmmurgiya@vbox:~/work/arch-pc/lab06$ nasm -f elf lab6-3.asm
mmmurgiya@vbox:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-3 lab6-3.o
mmmurgiya@vbox:~/work/arch-pc/lab06$ ./lab6-3
Результат: 4
Остаток от деления: 1
mmmurgiya@vbox:~/work/arch-pc/lab06$ nasm -f elf lab6-3.asm
mmmurgiya@vbox:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-3 lab6-3.o
mmmurgiya@vbox:~/work/arch-pc/lab06$ ./lab6-3
Результат: 5
Остаток от деления: 1
mmmurgiya@vbox:~/work/arch-pc/lab06$
```

Рис. 4.5: Результаты lab6-3.asm

Для последней задачи, я отвечу на вопросы связанные с lab6-4.asm.



```
touch variant.asm
mmmurgiya@vbox:~/work/arch-pc/lab06$ nasm -f elf variant.asm
mmmurgiya@vbox:~/work/arch-pc/lab06$ ld -m elf_i386 -o variant variant.o
mmmurgiya@vbox:~/work/arch-pc/lab06$ ./variant
Введите № студенческого билета:
1132243817
Ваш вариант: 18
mmmurgiya@vbox:~/work/arch-pc/lab06$
```

Рис. 4.6: lab6-4.asm работает

1. Данные строки отвечают за вывод на экран сообщения 'Ваш вариант':

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

2. Инструкция используется для ввода номера студенческого билета.
3. Переделывает данные нашего ввода в число, которое мы можем понять.
4. Ниже даны строки, подходящие к вопросу:

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

5. eax.
6. Добавляет единицу к регистру edx.
7. Вывод появляется следующими строками:

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

5 Выводы

Мы освоили арифметические инструкции языка ассемблера NASM.

Список литературы