

Лабораторная работа №6

Арифметические операции в NASM

Прозорова Елизавета Евгеньевна

Содержание

1	Цель работы	3
2	Выполнение лабораторной работы	4
3	Ответы на вопросы	12
4	Выполнение самостоятельной работы	13
5	Выводы	16

1 Цель работы

Освоение арифметических инструкций языка ассемблера NASM

2 Выполнение лабораторной работы

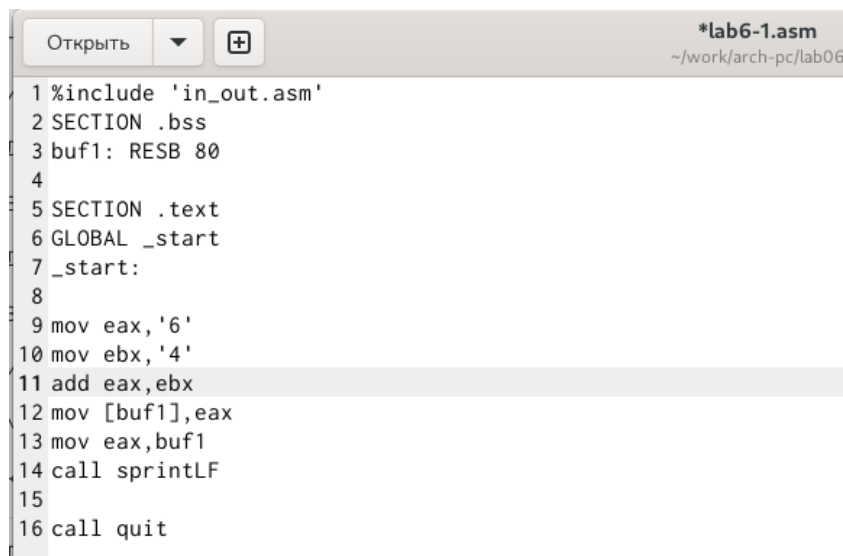
1. Сначала я создала каталог для программ лабораторной работы № 6, затем перешла в него и создала файл lab6-1.asm



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

Рис. 2.1: Создание каталога и файла lab6-1

2. Я ввела в файл lab6-1.asm текст программы из листинга 6.1.



```
*lab6-1.asm
~/work/arch-pc/lab06

1 %include 'in_out.asm'
2 SECTION .bss
3 buf1: RESB 80
4
5 SECTION .text
6 GLOBAL _start
7 _start:
8
9 mov eax, '6'
10 mov ebx, '4'
11 add eax, ebx
12 mov [buf1], eax
13 mov eax, buf1
14 call sprintLF
15
16 call quit
```

Рис. 2.2: Текст программы lab6-1

Я создала исполняемый файл и запустила его

```

eeprozorova@dk8n60 ~/work/arch-pc/lab06 $ nasm -f elf lab6-1.asm
eeprozorova@dk8n60 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-1 lab6-1.o
eeprozorova@dk8n60 ~/work/arch-pc/lab06 $ ./lab6-1
bash: ./lab6-1: Нет такого файла или каталога
eeprozorova@dk8n60 ~/work/arch-pc/lab06 $ ./lab6-1
j
eeprozorova@dk8n60 ~/work/arch-pc/lab06 $ 

```

Рис. 2.3: Создание и запуск lab6-1

3. Я изменила текст программы и вместо символов, запишем в регистры числа

```

SECTION .text
GLOBAL _start
_start:

mov eax,6
mov ebx,4
add eax,ebx

```

Рис. 2.4: Изменения текста в файле

Затем я создала и проверила измененный файл.

```

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

eeprozorova@dk8n60 ~/work/arch-pc/lab06 $ 

```

Рис. 2.5: Создание и запуск lab6-1

Коду 10 в таблице соответствует символ LF, при выводе на экран этот символ не отображается.

4. Я создала файл lab6-2.asm в каталоге ~/work/arch-pc/lab06 и ввела в него текст программы из листинга 6.2.

```

1 %include 'in_out.asm'
2 SECTION .text
3 GLOBAL _start
4 _start:
5
6 mov eax,'6'
7 mov ebx,'4'
8 add eax,ebx
9 call iprintLF
10
11 call quit

```

Рис. 2.6: Текст файла lab6-2

Затем я создала и проверила измененный файл.

```

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

```

Рис. 2.7: Создание и запуск lab6-2

5. Аналогично предыдущему пункту изменила символы на числа.

```

5
6 mov eax,6
7 mov ebx,4
8 add eax,ebx

```

Рис. 2.8: Изменения в тексте программы lab6-2

Затем я создала и проверила измененный файл.

```

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

```

Рис. 2.9: Создание и запуск lab6-2

В результате было получено число 10.

Затем я заменила функцию `iprintLF` на `iprint`. Создала исполняемый файл и запустила его

```
7 mov ebx,4
8 add eax,ebx
9 call iprint
0
1 call quit
```

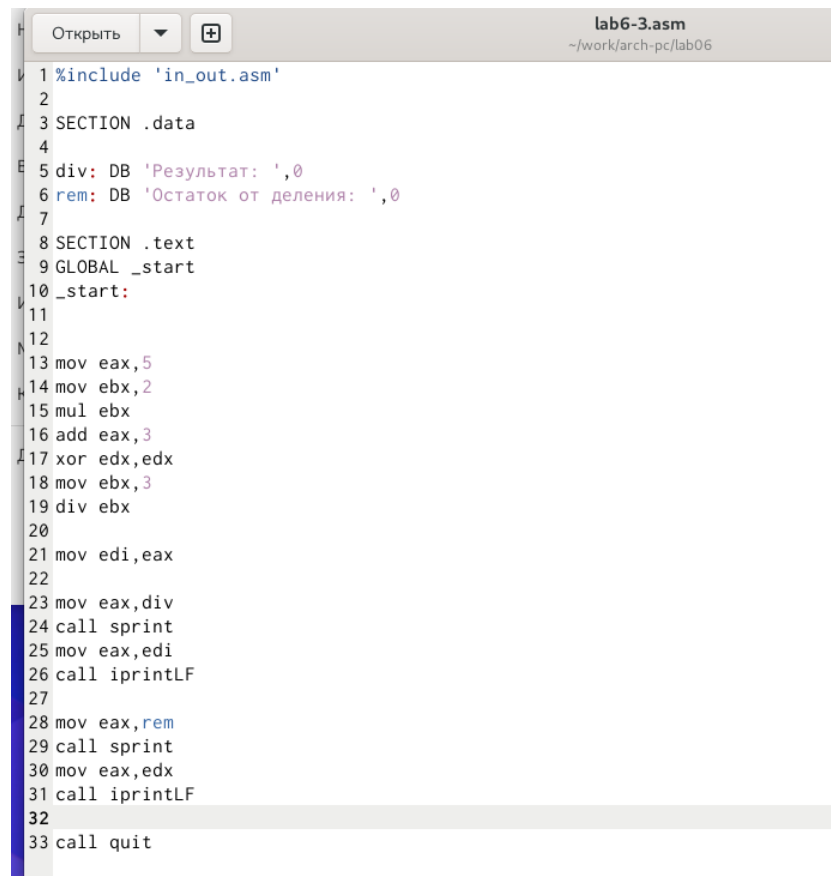
Рис. 2.10: Замена функции `iprintLF` на `iprint`

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

Рис. 2.11: Создание и запуск `lab6-2`

Вывод функций `iprintLF` и `iprint` отличается тем что, `iprint` не добавляет перенос курсора на новую строку

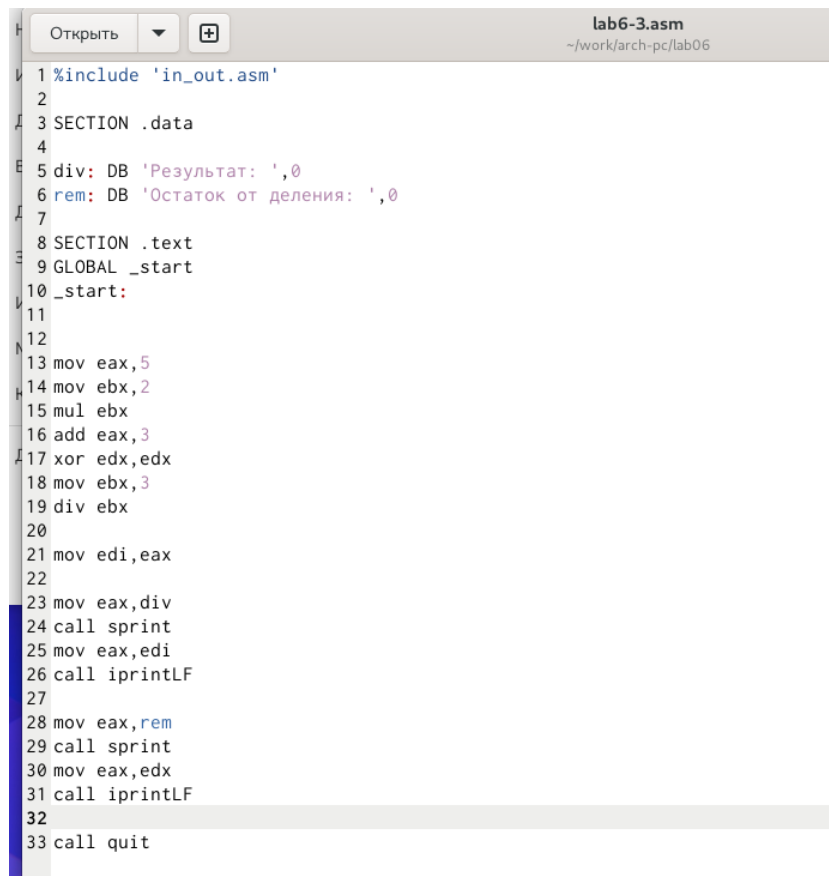
6. Я создала файл `lab6-3.asm` в каталоге `~/work/arch-pc/lab06`. Ввела текст программы из листинга 6.3 в `lab6-3.asm`



```
1 %include 'in_out.asm'
2
3 SECTION .data
4
5 div: DB 'Результат: ',0
6 rem: DB 'Остаток от деления: ',0
7
8 SECTION .text
9 GLOBAL _start
10 _start:
11
12
13 mov eax,5
14 mov ebx,2
15 mul ebx
16 add eax,3
17 xor edx,edx
18 mov ebx,3
19 div ebx
20
21 mov edi,eax
22
23 mov eax,div
24 call sprint
25 mov eax,edi
26 call iprintLF
27
28 mov eax,rem
29 call sprint
30 mov eax,edx
31 call iprintLF
32
33 call quit
```

Рис. 2.12: Текст программы lab6-3

Затем я создала и проверила измененный файл.



```
1 %include 'in_out.asm'
2
3 SECTION .data
4
5 div: DB 'Результат: ',0
6 rem: DB 'Остаток от деления: ',0
7
8 SECTION .text
9 GLOBAL _start
10 _start:
11
12
13 mov eax,5
14 mov ebx,2
15 mul ebx
16 add eax,3
17 xor edx,edx
18 mov ebx,3
19 div ebx
20
21 mov edi,eax
22
23 mov eax,div
24 call sprint
25 mov eax,edi
26 call iprintLF
27
28 mov eax,rem
29 call sprint
30 mov eax,edx
31 call iprintLF
32
33 call quit
```

Рис. 2.13: Создание и запуск lab6-3

Я изменила текст программы для вычисления выражения $\square(\square) = (4\square6 + 2)/5$



```
10 _start:
11
12
13 mov eax,4
14 mov ebx,6
15 mul ebx
16 add eax,2
17 xor edx,edx
18 mov ebx,5
19 div ebx
20
21 mov edi,eax
```

Рис. 2.14: Изменения текста lab6-3

Создала и проверила измененный файл.

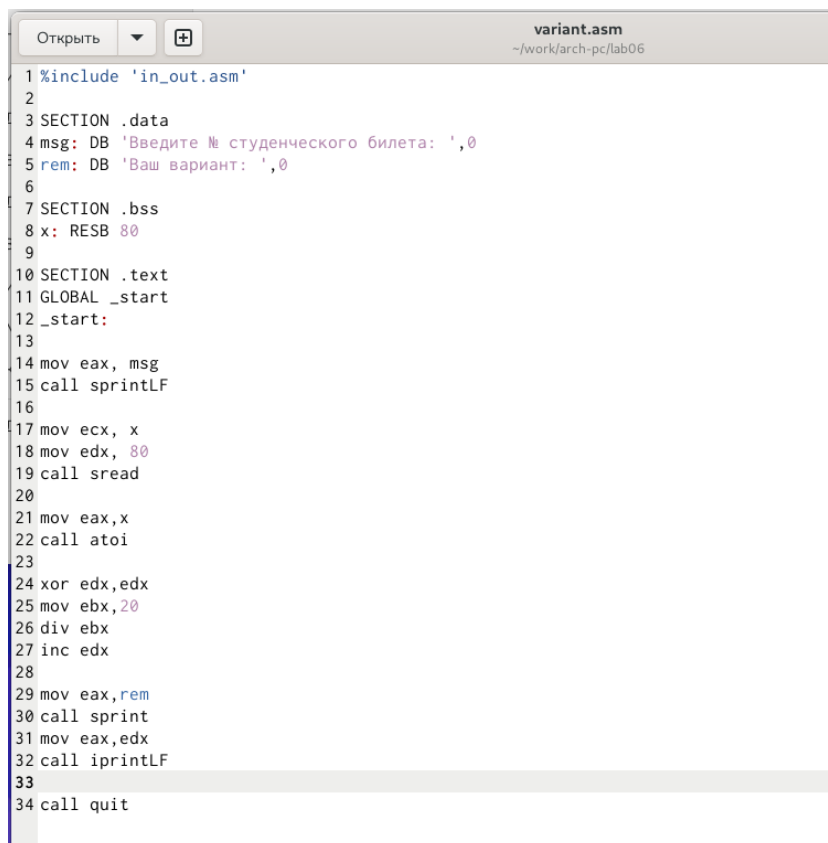
```

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

```

Рис. 2.15: Создание и запуск lab6-3

7. Я создайте файл `variant.asm` в каталоге `~/work/arch-pc/lab06`, в который ввела текст программы из листинга 6.4.



```

1 %include 'in_out.asm'
2
3 SECTION .data
4 msg: DB 'Введите № студенческого билета: ',0
5 rem: DB 'Ваш вариант: ',0
6
7 SECTION .bss
8 x: RESB 80
9
10 SECTION .text
11 GLOBAL _start
12 _start:
13
14 mov eax, msg
15 call sprintf
16
17 mov ecx, x
18 mov edx, 80
19 call sread
20
21 mov eax, x
22 call atoi
23
24 xor edx, edx
25 mov ebx, 20
26 div ebx
27 inc edx
28
29 mov eax, rem
30 call sprintf
31 mov eax, edx
32 call iprintLF
33
34 call quit

```

Рис. 2.16: Текст программы variant.asm

```

eeprozorova@dk8n60 ~/work/arch-pc/lab06 $ touch ~/work/arch-pc/lab06/variant.asm
eeprozorova@dk8n60 ~/work/arch-pc/lab06 $ nasm -f elf variant.asm
eeprozorova@dk8n60 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o variant variant.o
eeprozorova@dk8n60 ~/work/arch-pc/lab06 $ ./variant
Введите № студенческого билета:
1132246767
Ваш вариант: 8

```

Рис. 2.17: Создание и запуск variant.asm

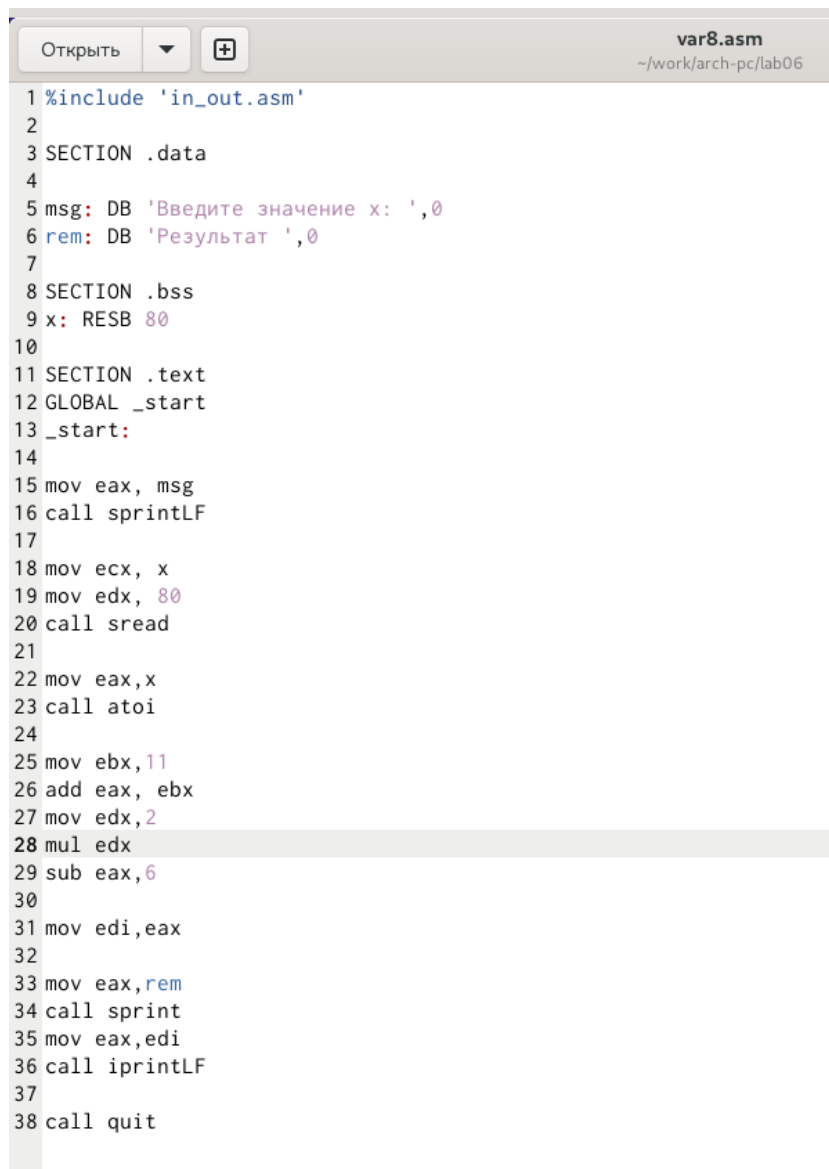
Мой вариант - 8.

3 Ответы на вопросы

- 1) `mov eax, msg call sprintLF`
- 2) `mov ecx, x` - устанавливает значение x в регистр `eax` `mov edx, 80` - команда устанавливает значение 80, длину вводимой строки
- 3) `call atoi` преобразует строку символов в целое число.
- 4) `xor edx,edx mov ebx,20 div ebx inc edx`
- 5) `ebx`
- 6) `inc` - прибавление 1 к операнду
- 7) `mov eax,rem call sprint mov eax,edx call iprintLF`

4 Выполнение самостоятельной работы

Я написала программу вычисления выражения $y = x(x)$, в моем варианте $x(x) = (11 + x)x^2 - 6$. Файл с текстом программы я назвала var8.asm



The screenshot shows a text editor window titled 'var8.asm' with the path '~/.work/arch-pc/lab06'. The code is an x86 assembly program. It starts with a line number 1 and includes 'in_out.asm'. It defines a data section with a message 'Введите значение x: ',0 and a result string 'Результат ',0. It then defines a bss section with a variable 'x' of size 80. The main code starts at line 12 with a global _start. It moves the address of 'msg' to 'eax', calls 'sprintf', moves the address of 'x' to 'ecx', moves the value 80 to 'edx', calls 'sread', moves the value from 'x' to 'eax', calls 'atoi', moves the value 11 to 'ebx', adds 'ebx' to 'eax', moves the value 2 to 'edx', multiplies 'edx' by 'eax' (line 28 is highlighted), subtracts 6 from 'eax', moves 'eax' to 'edi', moves the address of 'rem' to 'eax', calls 'sprintf', moves 'edi' to 'eax', calls 'iprintLF', and finally calls 'quit'.

```
1 %include 'in_out.asm'
2
3 SECTION .data
4
5 msg: DB 'Введите значение x: ',0
6 rem: DB 'Результат ',0
7
8 SECTION .bss
9 x: RESB 80
10
11
12 SECTION .text
13 GLOBAL _start
14 _start:
15 mov eax, msg
16 call sprintf
17
18 mov ecx, x
19 mov edx, 80
20 call sread
21
22 mov eax, x
23 call atoi
24
25 mov ebx, 11
26 add eax, ebx
27 mov edx, 2
28 mul edx
29 sub eax, 6
30
31 mov edi, eax
32
33 mov eax, rem
34 call sprintf
35 mov eax, edi
36 call iprintLF
37
38 call quit
```

Рис. 4.1: Готовая программа для варианта 8

Затем я несколько раз проверила исправность работы программы и исправила ошибки.

```
eeprozorova@dk3n55 ~/work/arch-pc/lab06 $ nasm -f elf var8.asm
eeprozorova@dk3n55 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o var8 var8.o
eeprozorova@dk3n55 ~/work/arch-pc/lab06 $ ./var8
Введите значение x:
1
Результат 18
eeprozorova@dk3n55 ~/work/arch-pc/lab06 $ ./var8
Введите значение x:
9
Результат 34
eeprozorova@dk3n55 ~/work/arch-pc/lab06 $ □
```

Рис. 4.2: Создание и запуск var8

5 Выводы

В ходе лабораторной работы были изучены арифметические инструкции языка ассемблера NASM.