

Отчет по лабораторной работе №6

Арифметические операции в NASM

Полина Алексеевна Ларионова

Содержание

1	Цель работы	4
2	Задание	5
3	Выполнение лабораторной работы	6
4	Ответы на вопросы	12
5	Задание для самостоятельной работы	13
6	Выводы	14

Список иллюстраций

3.1	Создание каталога	6
3.2	lab6-1.asm	6
3.3	Исполняемый файл 1	7
3.4	Регистры числа	7
3.5	Исполняемый файл 2	7
3.6	lab6-2.asm	8
3.7	Исполняемый файл 3	8
3.8	Символы изменены на числа	8
3.9	Исполняемый файл 4	9
3.10	iprint	9
3.11	lab6-3.asm	9
3.12	Исполняемый файл 5	10
3.13	Измененное выражение	10
3.14	variant.asm	11
3.15	Исполняемый файл 6	11
5.1	Вариант 10	13

1 Цель работы

Освоение арифметических инструкций языка ассемблера NASM.

2 Задание

Ознакомиться с основными арифметическими операциями на языке Ассемблера, символьными и численными данными. Выполнить арифметические операции и вариант вычисления задания по номеру студенческого билета.

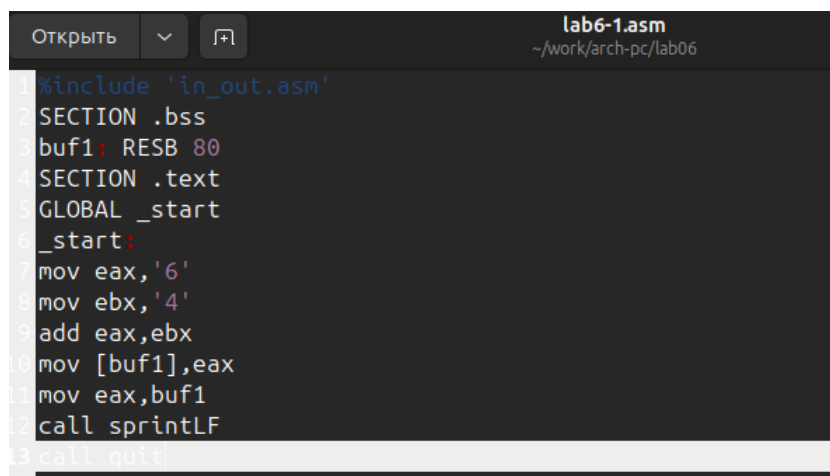
3 Выполнение лабораторной работы

Я создала каталог для программ лабораторной работы №6, создала файл lab6-1.asm и открыла его.

```
palarionova@linux:~$ mkdir ~/work/arch-pc/lab06
palarionova@linux:~$ cd ~/work/arch-pc/lab06
palarionova@linux:~/work/arch-pc/lab06$ touch lab6-1.asm
palarionova@linux:~/work/arch-pc/lab06$
```

Рис. 3.1: Создание каталога

В соответствии с листингом 6.1 я ввела в файл текст программы.



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

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

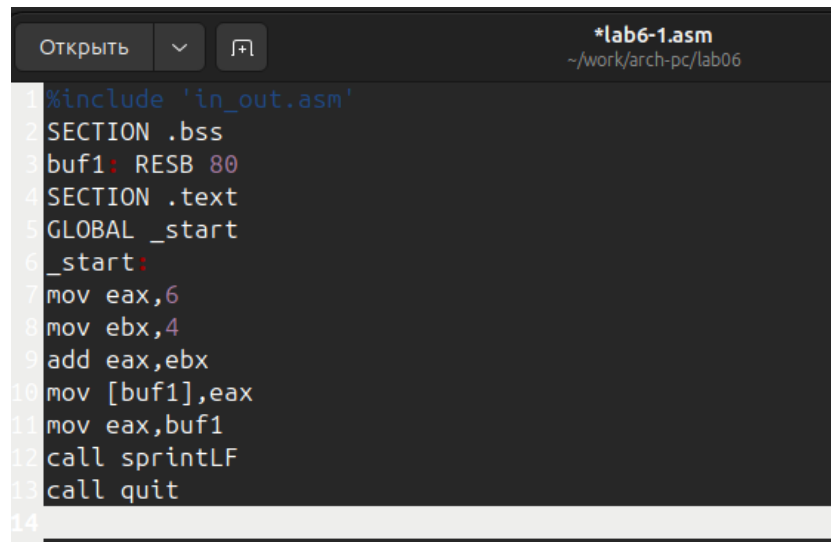
Рис. 3.2: lab6-1.asm

Далее я создала исполняемый файл и запустила его.

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

Рис. 3.3: Исполняемый файл 1

Затем я изменила текст программы, записав регистры числа вместо символов.



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

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

Рис. 3.4: Регистры числа

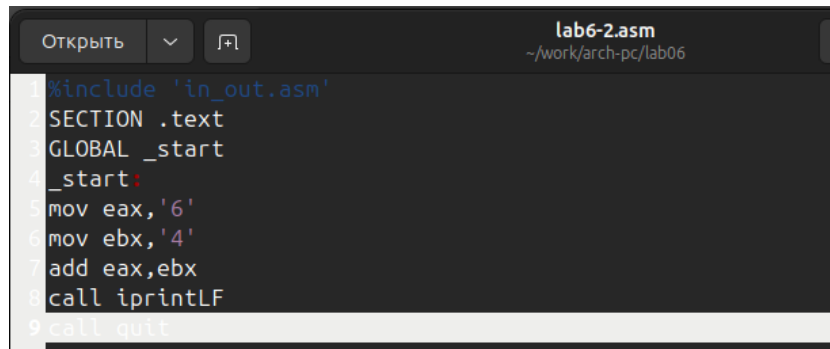
И так же создала исполняемый файл, в котором символ не был отображен при выводе на экран.

```
palarionova@linux:~/work/arch-pc/lab06$ gedit lab6-1.asm
palarionova@linux:~/work/arch-pc/lab06$ nasm -f elf lab6-1.asm
palarionova@linux:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-1 lab6-1.o
palarionova@linux:~/work/arch-pc/lab06$ ./lab6-1

palarionova@linux:~/work/arch-pc/lab06$
```

Рис. 3.5: Исполняемый файл 2

Далее я создала файл lab6-2.asm и ввела в него текст программы из листинга 6.2



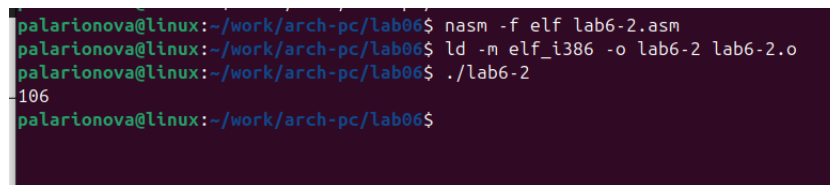
```

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

```

Рис. 3.6: lab6-2.asm

И создала исполняемый файл и запустила его.



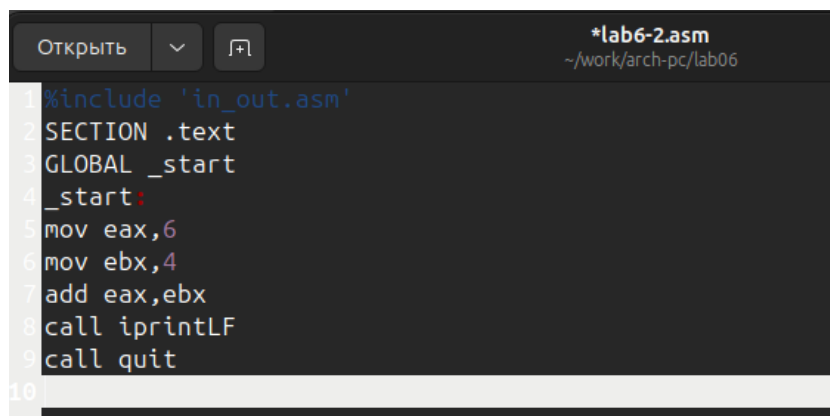
```

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

```

Рис. 3.7: Исполняемый файл 3

Аналогично предыдущему примеру я изменила символы на числа и запустила исполняемый файл.



```

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

```

Рис. 3.8: Символы изменены на числа


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

Рис. 3.9: Исполняемый файл 4

Затем я заменила функцию `iprintLF` на `iprint` и запустила исполняемый файл. Различие выводов функций заключается в отображении числа на разных строках.

```
palarionova@linux:~/work/arch-pc/lab06$ nasm -f elf lab6-2.asm
palarionova@linux:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-2 lab6-2.j
ld: невозможно найти lab6-2.j: Нет такого файла или каталога
palarionova@linux:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-2 lab6-2.o
palarionova@linux:~/work/arch-pc/lab06$ ./lab6-2
10
palarionova@linux:~/work/arch-pc/lab06$
```

Рис. 3.10: `iprint`

Создав файл `lab6-3.asm`, я ввела в файл текст программы из листинга 6.3

```
Открыть  ▾  [icon]  *lab6-3.asm
~/work/arch-pc/lab06

;-----
; Программа вычисления выражения
;-----
%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
call iprintLF ; из 'edi' в виде символов
mov eax,rem ; вызов подпрограммы печати
call sprint ; сообщения 'Остаток от деления: '
mov eax,edx ; вызов подпрограммы печати значения
call iprintLF ; из 'edx' (остаток) в виде символов
call quit ; вызов подпрограммы завершения
30
```

Рис. 3.11: `lab6-3.asm`

и проверила работу программы, создав исполняемый файл.

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

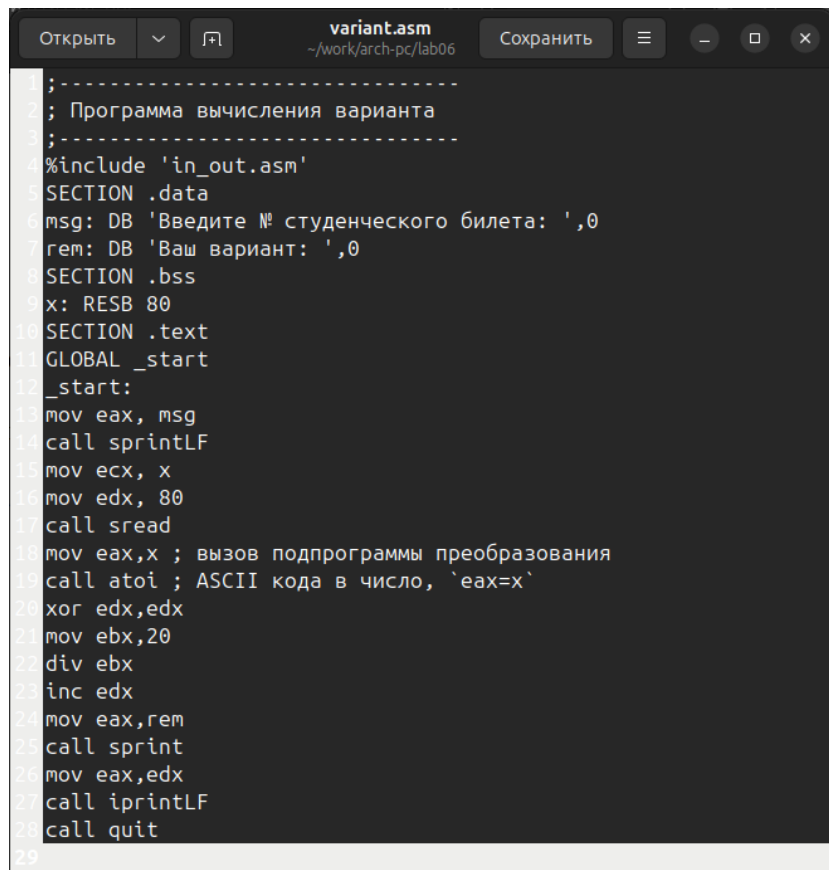
Рис. 3.12: Исполняемый файл 5

Затем я изменила текст программы для вычисления другого выражения и проверила работу.

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

Рис. 3.13: Измененное выражение

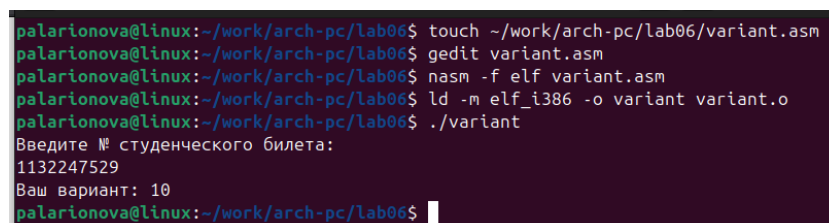
Далее я создала файл variant.asm и ввела в него текст программы из листинга 6.4



```
1;-----
2; Программа вычисления варианта
3;-----
4#include 'in_out.asm'
5SECTION .data
6msg: DB 'Введите № студенческого билета: ',0
7rem: DB 'Ваш вариант: ',0
8SECTION .bss
9x: RESB 80
10SECTION .text
11GLOBAL _start
12_start:
13mov eax, msg
14call sprintf
15mov ecx, x
16mov edx, 80
17call sread
18mov eax,x ; вызов подпрограммы преобразования
19call atoi ; ASCII кода в число, `eax=x`
20xor edx,edx
21mov ebx,20
22div ebx
23inc edx
24mov eax,rem
25call sprintf
26mov eax,edx
27call iprintLF
28call quit
29
```

Рис. 3.14: variant.asm

и проверила работу программы, узнав свой вариант для выполнения самостоятельной работы.



```
palarionova@linux:~/work/arch-pc/lab06$ touch ~/work/arch-pc/lab06/variant.asm
palarionova@linux:~/work/arch-pc/lab06$ gedit variant.asm
palarionova@linux:~/work/arch-pc/lab06$ nasm -f elf variant.asm
palarionova@linux:~/work/arch-pc/lab06$ ld -m elf_i386 -o variant variant.o
palarionova@linux:~/work/arch-pc/lab06$ ./variant
Введите № студенческого билета:
1132247529
Ваш вариант: 10
palarionova@linux:~/work/arch-pc/lab06$
```

Рис. 3.15: Исполняемый файл 6

4 Ответы на вопросы

1) Какие строки листинга 6.4 отвечают за вывод на экран сообщения 'Ваш вариант: '? `mov eax,msg call sprintLF`

2) Для чего используются следующие инструкции? `mov ecx, x mov edx, 80 call sread`

Они используются для ввода переменной X с клавиатуры и сохранения введенных данных.

3) Для чего используется инструкция "call atoi"?

Для преобразования кода переменной ASCII в число.

4) Какие строки листинга 6.4 отвечают за вычисления варианта?

`mov ebx,20 div ebx inc edx`

5) В какой регистр записывается остаток от деления при выполнении инструкции "div ebx"?

В регистр ebx.

6) Для чего используется инструкция "inc edx"?

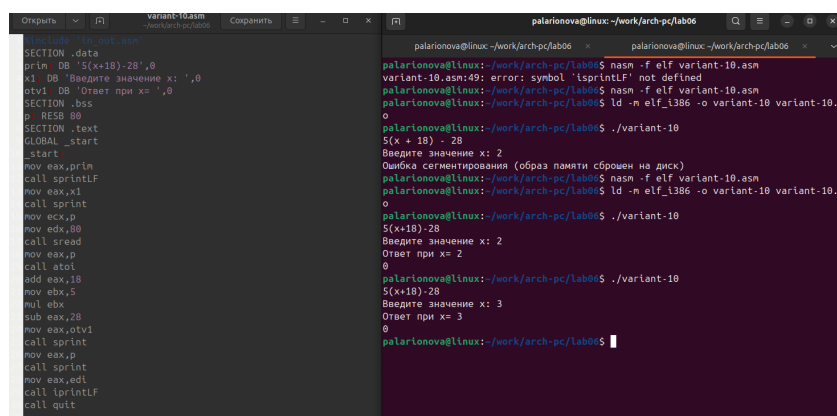
Для увеличения edx на 1.

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

`mov eax, edx call iprintLF`

5 Задание для самостоятельной работы

Получив свой вариант, я написала программу для вычисления выражения из варианта 10, но при запуске исполняемого файла получала неверные ответы для x_1 и x_2 .



```
variant-10.asm
SECTION .data
prln DB '5(x+18)-28',0
x1 DB 'Введите значение x: ',0
otvi DB 'Ответ при x= ',0
SECTION .bss
p RESB 80
SECTION .text
GLOBAL _start
_start:
mov eax,prln
call sprintf
mov eax,x1
call sprintf
mov ecx,p
mov edx,80
call sread
mov eax,p
call atoi
add eax,18
mov ebx,5
mul ebx
sub eax,28
mov eax,otvi
call sprintf
mov eax,p
call sprintf
mov eax,edi
call sprintf
call quit

palarionova@linux:~/work/arch-pc/lab06$ nasm -f elf variant-10.asm
variant-10.asm:49: error: symbol 'sprintf' not defined
palarionova@linux:~/work/arch-pc/lab06$ nasm -f elf variant-10.asm
palarionova@linux:~/work/arch-pc/lab06$ ld -m elf_i386 -o variant-10 variant-10.o
palarionova@linux:~/work/arch-pc/lab06$ ./variant-10
5(x + 18) - 28
Введите значение x: 2
Ошибка сегментирования (образ памяти сброшен на диск)
palarionova@linux:~/work/arch-pc/lab06$ nasm -f elf variant-10.asm
palarionova@linux:~/work/arch-pc/lab06$ ld -m elf_i386 -o variant-10 variant-10.o
palarionova@linux:~/work/arch-pc/lab06$ ./variant-10
5(x+18)-28
Введите значение x: 2
Ответ при x= 2
0
palarionova@linux:~/work/arch-pc/lab06$ ./variant-10
5(x+18)-28
Введите значение x: 3
Ответ при x= 3
0
palarionova@linux:~/work/arch-pc/lab06$
```

Рис. 5.1: Вариант 10

6 Выводы

Я освоила арифметические операции на языке Ассемблера.