

**Российский университет дружбы народов имени  
Патриса Лумумбы**

**Отчёт по лабораторной работе №6**

**По теме: Арифметические операции в NASM.**

Выполнил: Фомин Виктор Владимирович, НММбд-04-24

**Содержание**

Цель работы.....	1
Ход выполнения лабораторной работы.....	2
Выполнение самостоятельной работы.....	
Вывод.....	9
Список	
литературы.....	10

**Цель работы:** Освоение арифметических инструкций языка  
ассемблера NASM

**Ход выполнения лабораторной работы.**

Я создал каталог lab06 и файл lab6-1.asm.

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

Далее перешел в созданный файл и ввёл текст программы из листинга 6.1. Программа должна выводить значения, записанные в регистр eax:

```
GNU nano 7.2 /home/vvfomin/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 sprintLF
call quit
```

Создал исполняемый файл и запустил его:

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

При выводе значения регистра `eax` мы ожидаем увидеть число 10. Однако результатом был символ `j`. Это произошло потому, что код символа `6` равен 00110110 в двоичном представлении (или 54 в десятичном представлении), а код символа `4` – 00110100, что в свою очередь является кодом символа `j`.

Далее я изменил текст программы, и вместо символов я записал в регистры числа.

```
%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
```

Создал исполняемый файл и запустил его

```

vvfomin@fominvv:~/work/arch-pc/lab06$ nasm -f elf lab6-1.asm
vvfomin@fominvv:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-1 lab6-1.o
ld: cannot find lab6-1.o: No such file or directory
vvfomin@fominvv:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-1 lab6-1.o
vvfomin@fominvv:~/work/arch-pc/lab06$ ./lab6-1

vvfomin@fominvv:~/work/arch-pc/lab06$

```

Как и в предыдущем примере число 10 не вывелось в терминале. Вывелся символ с кодом 10. Пользуясь таблицей ASCII из приложения 2 я определил, что число 10 равно символу “LF,\n”. Этот символ не отображается при выводе на экран.:

Я создал файл lab6-2.asm и ввел в него текст программы

in_out.asm	3942	Feb 21 13:38	in_out.asm	3942	Feb 21 13:38
*lab6-1	5160	Feb 22 19:40	*lab6-1	5160	Feb 22 19:40
lab6-1.asm	169	Feb 22 19:39	lab6-1.asm	169	Feb 22 19:39
lab6-1.o	1200	Feb 22 19:40	lab6-1.o	1200	Feb 22 19:40
lab6-2.asm	169	Feb 22 19:39	lab6-2.asm	169	Feb 22 19:39
lab6-2.asm			UP - - DIR		
12G / 49G (23%)			12G / 49G (23%)		

```
%include 'in_out.asm'
SECTION .text
GLOBAL _start
_start:
mov eax,'6'
mov ebx,'4'
add eax,ebx
call iprintLF
call quit
```

В результате работы программы я получил число 106. В данном случае, как и в первом, команда add складывает коды символов '6' и '4' (54+52=106)

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

Аналогично заменил символы на числа.

```
%include 'in_out.asm'
SECTION .text
GLOBAL _start
_start:
mov eax,6
mov ebx,4
add eax,ebx
call iprintLF
call quit
```

```

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

```

Я заменил функцию `iprintLF` на `iprint`. Создал исполняемый файл и запустил его. Отличие в том, что результат вывелся на одной строке вместе со строкой `vvfomin@fominvv`:

```

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

```

Я создал файл `lab6-3.asm`. Далее с помощью листинга 6.3 я написал программу вычисления арифметического выражения  $f(x)=(5*2+3)/3$ :

in_out.asm	3942	Feb 21 13:38	in_out.asm	3942	Feb 21 13:38
*lab6-1	5160	Feb 22 19:40	*lab6-1	5160	Feb 22 19:40
lab6-1.asm	169	Feb 22 19:39	lab6-1.asm	169	Feb 22 19:39
lab6-1.o	1200	Feb 22 19:40	lab6-1.o	1200	Feb 22 19:40
*lab6-2	5088	Feb 23 08:28	*lab6-2	5088	Feb 23 08:28
lab6-2.asm	112	Feb 23 08:27	lab6-2.asm	112	Feb 23 08:27
lab6-2.o	1040	Feb 23 08:27	lab6-2.o	1040	Feb 23 08:27
lab6-3.asm	112	Feb 23 08:27	lab6-3.asm	112	Feb 23 08:27

```

%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
div ebx
mov edi,eax

mov eax,div
call sprint
mov eax,edi
call iprintLF

```

Далее создал исполняемый файл и запустил его. Результат работы получился следующим:

```

vvfomin@fominvv:~/work/arch-pc/lab06$ nasm -f elf lab6-3.asm
vvfomin@fominvv:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-3 lab6-3.o
vvfomin@fominvv:~/work/arch-pc/lab06$ ./lab6-3
Результат: 6
Остаток от деления: 1
vvfomin@fominvv:~/work/arch-pc/lab06$

```

Аналогично я изменил текст программы для вычисления выражения  $f(x)=(4*6+2)/5$ .

```

#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

```

```

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

```

Я создал файл var для написания программы, которая вычисляет вариант задания по номеру студенческого билета. Далее я написал код самой программы с помощью листинга 6.4:



```

%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 sprintf
mov ecx, x
mov edx, 80
call sread
mov eax, x ; вызов подпрограммы преобразования
call atoi
xor edx, edx
mov ebx, 20
div ebx
inc edx

```

Я создал исполняемый файл и запустил его. В результате в терминале вывелось следующее:

```

vvfomin@fominvv:~/work/arch-pc/lab06$ nasm -f elf var.asm
vvfomin@fominvv:~/work/arch-pc/lab06$ ld -m elf_i386 -o var var.o
vvfomin@fominvv:~/work/arch-pc/lab06$ ./var
Введите № студенческого билета:
1132247528
Ваш вариант: 9
vvfomin@fominvv:~/work/arch-pc/lab06$

```

## Вывод:

В ходе данной лабораторной работы я освоил полезные арифметические инструкции языка ассемблера NASM.

## 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/Learningbash-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](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).