

# Отчет по лабораторной работа №6

## Дисциплина: архитектура компьютера

Ширинкин Т. Б.

### Содержание

Цель работы .....	1
Выполнение лабораторной работы .....	1
Выполнение заданий для самостоятельной работы.....	7
Выводы .....	9

### Цель работы

Приобретение навыков работы с арифметическими инструкциями языка ассемблер.

### Выполнение лабораторной работы

Создал каталог и первый файл (Рис. [-@fig:000])

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

*Рис. 0 Создал каталог и первый файл*

Ввёл первый листинг (Рис. [-@fig:001])

~ ~ ~ ~ ~

Рис. 1 Ввёл первый листинг

Создал исполняемый файл и запустил его (Рис. [-@fig:002])

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

*Рис. 2 Создал исполняемый файл и запустил его*

Проверяем, что он и правда создан(плюс показываю, что поместил в папку in\_out.asm)  
(Рис. [-@fig:003])

```
tbshirinkin@Shiza:~/work/arch-pc/lab06$ ls
in_out.asm  lab6-1    lab6-1.asm  lab6-1.o
```

Рис. 3 Проверяем, что он и правда создан(плюс показываю, что поместил в папку in\_out.asm)

### Заменяем символы цифр на цифры (Рис. [-@fig:004])

```

#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

-- INSERT --
8,10 All

```

Рис. 4 Заменяем символы цифр на цифры

Заработало: 10 - код символа переноса на новую строку (Рис. [-@fig:005])

```

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

```

Рис. 5 Заработало: 10 - код символа переноса на новую строку

Создал второй файл (Рис. [-@fig:006])

```

tbshirinkin@Shiza:~/work/arch-pc/lab06$ touch lab6-2.asm

```

Рис. 6 Создал второй файл

Ввёл в него второй код (Рис. [-@fig:007])



Вводим очередной листинг (Рис. [-@fig:012])

```
-----
; 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
|
~
~
~
~
~
~
~
~
30,0-1 All
```

Рис. 12 Вводим очередной листинг

Проверяем: да ответ предсказываемый (Рис. [-@fig:013])

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

Рис. 13 Проверяем: да ответ предсказываемый

Собрал, проверил - работает. (Рис. [-@fig:014])

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

Рис. 14 Собрал, проверил - работает.

Создаю новый файл (Рис. [-@fig:015])

```
tbshirinkin@Shiza:~/work/arch-pc/lab06$ touch variant.asm
```

Рис. 15 Создаю новый файл

Ввожу очередной листинг (Рис. [-@fig:016])



## Выполнение заданий для самостоятельной работы

Пишу код под свой номер ( $1 : 3 = 0$ , после чего ноль умножается на вводимый аргумент, поэтому ответ одинаков -  $5 * 7 = 35$ ) (Рис. [-@fig:018])

```
;-----  
; Программа самостоятельная  
;-----  
%include 'in_out.asm'  
SECTION .data  
msg: DB 'Введите x: ',0  
rem: DB 'Ваш f(x): ',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  
call atoi  
  
xor edx, edx  
mov eax, 1  
mov ebx, 3  
div ebx  
mov ebx, edx  
mul ebx  
add eax, 5  
mov ebx, 7  
mul ebx  
mov edx, eax  
  
mov eax, rem  
call sprint  
mov eax, edx  
call iprintLF  
call quit
```



Рис. 18 Пишу код под свой номер ( $1 : 3 = 0$ , после чего ноль умножается на вводимый аргумент, поэтому ответ одинаков -  $5 * 7 = 35$ ).

Проверил  $x1$  и  $x2 : 35$  и  $35$  (Рис. [-@fig:019])

```
tbshirinkin@Shiza:~/work/arch-pc/lab06$ nasm -f elf sm-19-lab6.asm
tbshirinkin@Shiza:~/work/arch-pc/lab06$ ld -m elf_i386 -o sm-19-lab6 sm-19-lab6.o
tbshirinkin@Shiza:~/work/arch-pc/lab06$ ./sm-19-lab6
Введите x:
3
Ваш f(x): 35
tbshirinkin@Shiza:~/work/arch-pc/lab06$ ./sm-19-lab6
Введите x:
9
Ваш f(x): 35
```

Рис. 19 Проверил  $x1$  и  $x2 : 35$  и  $35$

## Выводы

Цель достигнута: приобретены навыки работы с арифметическими инструкциями языка ассемблер.