

# **Отчёт по лабораторной работе №6**

**Архитектура компьютера**

Сафиуллина Айлина Саяровна

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Задание</b>	<b>6</b>
<b>3</b>	<b>Выполнение лабораторной работы</b>	<b>7</b>
<b>4</b>	<b>Выводы</b>	<b>16</b>

# Список иллюстраций

3.1	создание каталога . . . . .	7
3.2	листинг 6.1 . . . . .	7
3.3	создание исполняемого файла . . . . .	8
3.4	изменение текста программы . . . . .	8
3.5	создание исполняемого файла . . . . .	9
3.6	листинг 6.2 . . . . .	9
3.7	создание исполняемого файла . . . . .	9
3.8	создание исполняемого файла . . . . .	10
3.9	изменения в тексте программы . . . . .	10
3.10	создание исполняемого файла . . . . .	10
3.11	листинг 6.3 . . . . .	11
3.12	Создание исполняемого файла . . . . .	11
3.13	изменение текста программы . . . . .	12
3.14	создание исполняемого файла . . . . .	12
3.15	программа для вычисления варианта . . . . .	13
3.16	Создание исполняемого файла . . . . .	13
3.17	результат программы . . . . .	14
3.18	текст программы для вычисления выражения самостоятельной работы . . . . .	14

## **Список таблиц**

# 1 Цель работы

Целью работы является освоение арифметических инструкций языка ассемблера NASM.

## **2 Задание**

1. Символьные и численные данные в NASM.
2. Выполнение арифметических операций в NASM.
3. Выполнение задания для самостоятельной работы.

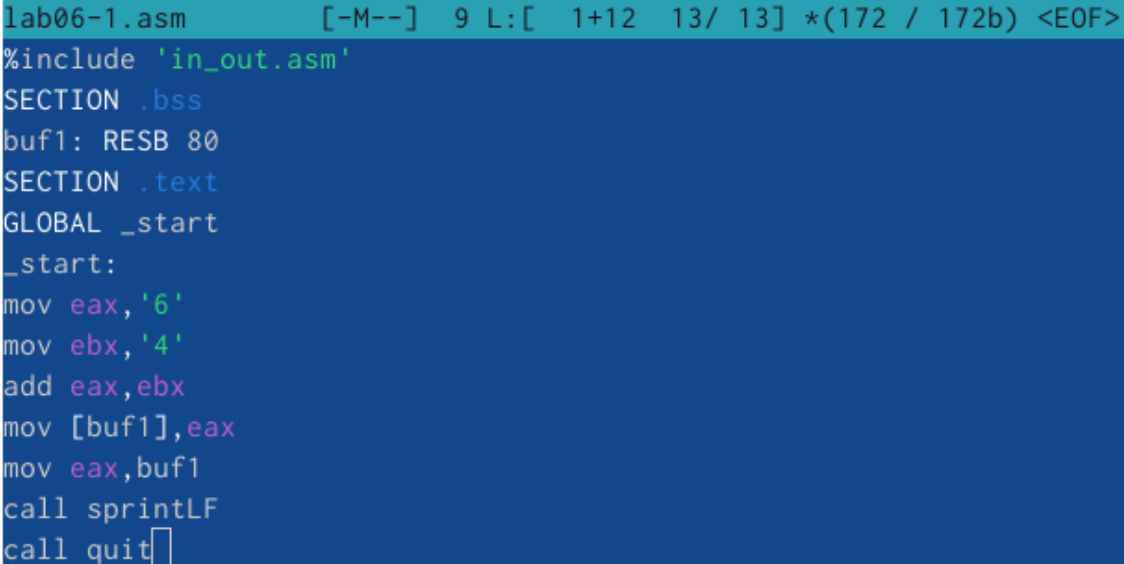
### 3 Выполнение лабораторной работы

Я создала каталог для программ лабораторной работы № 6, перешла в него и создала файл lab6-1.asm. (рис. 3.1).

```
assafiullina@dk1n22 ~ $ cd work/arch-pc
assafiullina@dk1n22 ~/work/arch-pc $ mkdir lab06
mkdir: невозможно создать каталог «lab06»: Файл существует
assafiullina@dk1n22 ~/work/arch-pc $ cp lab05/in_out.asm lab06/
assafiullina@dk1n22 ~/work/arch-pc $ cd lab06
assafiullina@dk1n22 ~/work/arch-pc/lab06 $ touch lab06-1.asm
assafiullina@dk1n22 ~/work/arch-pc/lab06 $ ls
```

Рис. 3.1: создание каталога

Вводим в файл lab6-1.asm текст программы из листинга 6.1 (рис. 3.2).



```
lab06-1.asm      [-M--]  9 L:[  1+12  13/ 13] *(172 / 172b) <EOF>
#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 sprintfLF
call quit
```

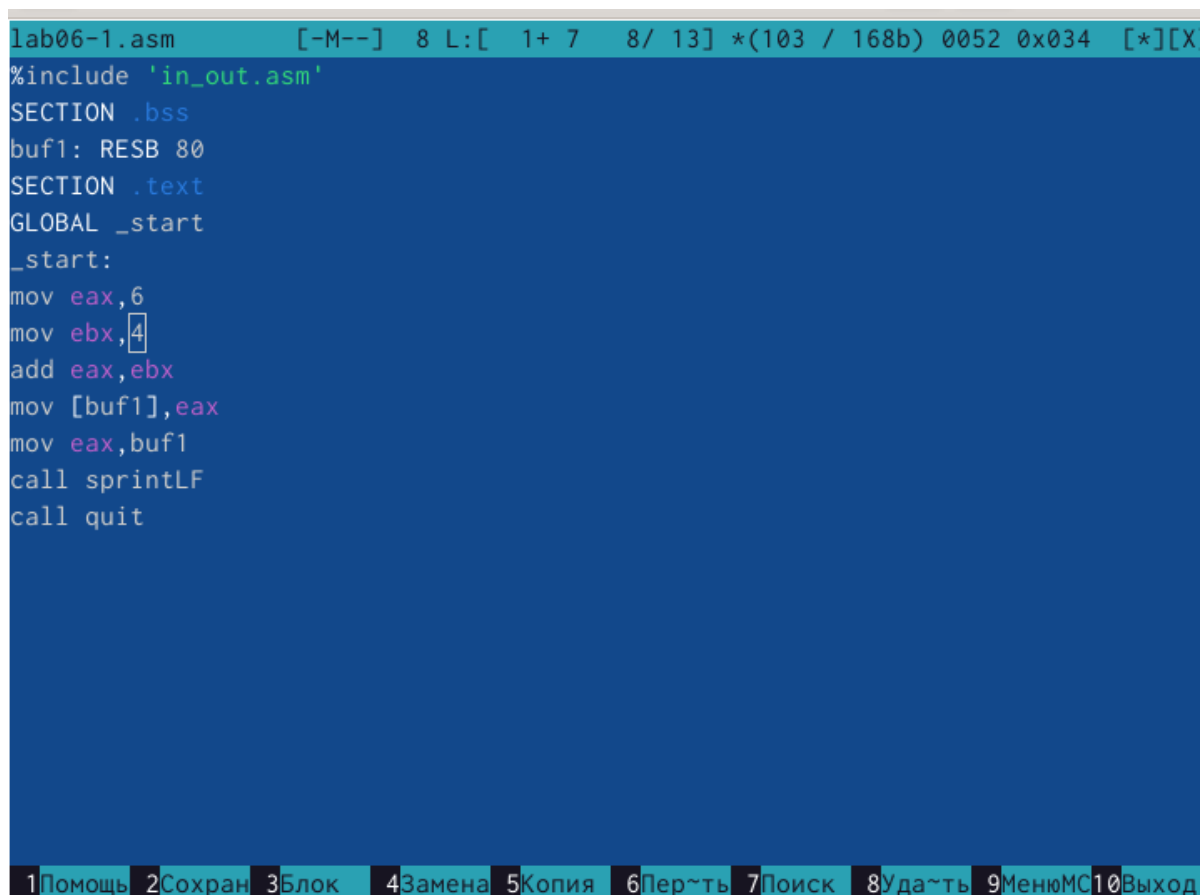
Рис. 3.2: листинг 6.1

Создадим исполняемый файл и проверим его работу (рис. 3.3).

```
assafiullina@dk1n22 ~/work/arch-pc/lab06 $ nasm -f elf lab06-1.asm
assafiullina@dk1n22 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab06-1 lab06-1.o
assafiullina@dk1n22 ~/work/arch-pc/lab06 $ ./lab06-1
j
assafiullina@dk1n22 ~/work/arch-pc/lab06 $ █
```

Рис. 3.3: создание исполняемого файла

Изменяю текст программы, вместо символов записывая в eax, ebx числа (рис. 3.4).



```
lab06-1.asm      [-M--]  8 L:[  1+ 7   8/ 13] *(103 / 168b) 0052 0x034  [*][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
```

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

Рис. 3.4: изменение текста программы

Создадим исполняемый файл и проверим его работу (рис. 3.5).



```

assafiullina@dk1n22 ~/work/arch-pc/lab06 $ nasm -f elf lab06-1.asm
assafiullina@dk1n22 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab06-1 lab06-1.o
assafiullina@dk1n22 ~/work/arch-pc/lab06 $ ./lab06-1

assafiullina@dk1n22 ~/work/arch-pc/lab06 $ █

```

Рис. 3.5: создание исполняемого файла

На экране ничего не отображается. Это связано с тем, что символ с кодом 10 - это символ перевода строки

Создаю файл lab6-2.asm в каталоге для программ лабораторной №6. Ввожу в него текст программы из листинга 6.2 (рис. 3.6).

```

lab06-2.asm      [-M--]  9 L:[ 1+ 8  9/  9] *(117 / 117b) <EOF>      [*][X]
#include 'in_out.asm'
SECTION .text
GLOBAL _start
_start:
mov eax,'6'
mov ebx,'4'
add eax,ebx
call iprintLF
call quit█

```

Рис. 3.6: листинг 6.2

Создадим исполняемый файл и проверим его работу (рис. 3.7).

```

assafiullina@dk1n22 ~/work/arch-pc/lab06 $ nasm -f elf lab06-2.asm
assafiullina@dk1n22 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab06-2 lab06-2.o
assafiullina@dk1n22 ~/work/arch-pc/lab06 $ ./lab06-2
106
assafiullina@dk1n22 ~/work/arch-pc/lab06 $ █

```

Рис. 3.7: создание исполняемого файла

Аналогично предыдущей программе заменяю символы на числа, затем снова создаю исполняемый файл и получаю результат арифметической операции (рис. 3.8).

```

assafiullina@dk1n22 ~/work/arch-pc/lab06 $ nasm -f elf lab06-2.asm
assafiullina@dk1n22 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab06-2 lab06-2.o
assafiullina@dk1n22 ~/work/arch-pc/lab06 $ ./lab06-2
10
assafiullina@dk1n22 ~/work/arch-pc/lab06 $ █

```

Рис. 3.8: создание исполняемого файла

Заменяю функцию `iprintLF` на `iprint` (рис. 3.9).

```

lab06-2.asm [-M--] 11 L:[ 1+ 7 8/ 9] *(101 / 111b) 0010 0x00A [*][X]
%include 'in_out.asm'
SECTION .text
GLOBAL _start
_start:
mov eax,6
mov ebx,4
add eax,ebx
call iprint█
call quit

```

Рис. 3.9: изменения в тексте программы

Создадим исполняемый файл и проверим его работу (рис. 3.10).

```

assafiullina@dk1n22 ~/work/arch-pc/lab06 $ nasm -f elf lab06-2.asm
assafiullina@dk1n22 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab06-2 lab06-2.o
assafiullina@dk1n22 ~/work/arch-pc/lab06 $ ./lab06-2
10assafiullina@dk1n22 ~/work/arch-pc/lab06 $ █

```

Рис. 3.10: создание исполняемого файла

Вывод функции `iprintLF` от вывода функции `iprint` отличается тем, что в последнем случае после вывода не добавляется переход на новую строку

С помощью функции `touch` создаю файл `lab6-3.asm`. Ввожу в него текст программы для вычисления значения указанного выражения (рис. 3.11).

```

lab6-3.asm      [----] 13 L:[ 3+ 3 6/ 26] *(198 /1236b) 0010 0x00A [*][X]
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
1Помощь 2Сохран 3Блок 4Замена 5Копия 6Переть 7Поиск 8Удасть 9МенюМС10Выход

```

Рис. 3.11: листинг 6.3

Создадим исполняемый файл и проверим его работу (рис. 3.12).

```

assafiullina@dk1n22 ~/work/arch-pc/lab06 $ nasm -f elf lab6-3.asm
assafiullina@dk1n22 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-3 lab6-3.o
assafiullina@dk1n22 ~/work/arch-pc/lab06 $ ./lab6-3
Результат: 4
Остаток от деления: 1

```

Рис. 3.12: Создание исполняемого файла

Изменяю текст программы для вычисления нового выражения (рис. 3.13).

```

lab6-3.asm      [-M--]  9 L:[ 5+21 26/ 26] *(1177/1236b) 0032 0x020 [*][
SECTION .text
GLOBAL _start
_start:
; ---- Вычисление выражения
mov eax,4 ; EAX=5
mov ebx,6 ; EBX=2
mul ebx ; EAX=EAX*EBX
add eax,2 ; EAX=EAX+3
xor edx,edx ; обнуляем EDX для корректной работы div
mov ebx,5 ; 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 iprintfLF ; 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 iprintfLF ; из 'edx' (остаток) в виде символов
call quit ; вызов подпрограммы завершения
1Помощь 2Сохран 3Блок 4Замена 5Копия 6Пер~ть 7Поиск 8Уда~ть 9МенюМС10Выхо

```

Рис. 3.13: изменение текста программы

Создадим исполняемый файл и проверим его работу (рис. 3.14).

```

assafiullina@dk1n22 ~/work/arch-pc/lab06 $ nasm -f elf lab6-3.asm
assafiullina@dk1n22 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-3 lab6-3.o
assafiullina@dk1n22 ~/work/arch-pc/lab06 $ ./lab6-3
Результат: 5
Остаток от деления: 1
assafiullina@dk1n22 ~/work/arch-pc/lab06 $ 

```

Рис. 3.14: создание исполняемого файла

С помощью функции touch создаю файл variant.asm. Ввожу в него текст программы для вычисления варианта (рис. 3.15).

```

variant.asm      [-M--] 25 L:[ 2+ 2  4/ 27] *(144 / 549b) 0010 0x00A  [*][>
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 ; ASCII кода в число, 'eax=x'
68 Демидова А. В.
Архитектура ЭВМ
xor edx, edx
mov ebx, 20
div ebx
inc edx
mov eax, rem

```

Рис. 3.15: программа для вычисления варианта

Создадим исполняемый файл и проверим его работу (рис. 3.16).

```

assafiullina@dk1n22 ~/work/arch-pc/lab06 $ nasm -f elf variant.asm
assafiullina@dk1n22 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o variant variant.o
assafiullina@dk1n22 ~/work/arch-pc/lab06 $ ./variant
Введите № студенческого билета:
1032241171
Ваш вариант: 12
assafiullina@dk1n22 ~/work/arch-pc/lab06 $ 

```

Рис. 3.16: Создание исполняемого файла

мой вариант - 12.

В соответствии с вариантом выполним задание для самостоятельной работы (рис. 3.17).

```

assafiullina@dk1n22 ~/work/arch-pc/lab06 $ nasm -f elf lab6-4.asm
assafiullina@dk1n22 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-4 lab6-4.o
assafiullina@dk1n22 ~/work/arch-pc/lab06 $ ./lab6-4
Введите X
1
Ошибка сегментирования (образ памяти сброшен на диск)

```

Рис. 3.17: результат программы

(рис. 3.18).

```

lab6-4.asm      [----] 10 L:[ 2+19 21/ 23] *(284 / 303b) 0120 0x078 [*][X]
SECTION .data
msg: DB 'Введите X ',0
rem: DB 'Выражение = : ',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

add eax,-6
mov ebx,18
mul ebx
xor edx,edx
mov ebx,2
div ebx

```

Рис. 3.18: текст программы для вычисления выражения самостоятельной работы

Ответы на вопросы: 1. Какие строки листинга отвечают за вывод на экран сообщения ‘Ваш вариант:’? • Строка “mov eax, rem” перекладывает в регистр значение переменной с фразой “Ваш вариант:”. Строка “call sprint” вызывает подпрограмму вывода строки. 2. Для чего используются следующие инструкции?

• Инструкция `"nasm"` используется для компиляции кода на языке ассемблера NASM. Инструкция `"mov esx, x"` используется для перемещения значения переменной `x` в регистр `esx`. Инструкция `"mov edx, 80"` используется для перемещения значения `80` в регистр `edx`. Инструкция `"call sread"` вызывает подпрограмму для считывания значения студенческого билета из консоли. 3. Для чего используется инструкция `"call atoi"`? • Инструкция `"call atoi"` используется для преобразования введенных символов в числовой формат. 4. Какие строки листинга отвечают за вычисления варианта? • Строка `"xor edx, edx"` обнуляет регистр `edx`. Строка `"mov ebx, 20"` записывает значение `20` в регистр `ebx`. Строка `"div ebx"` выполняет деление номера студенческого билета на `20`. Строка `"inc edx"` увеличивает значение регистра `edx` на `1`. 5. В какой регистр записывается остаток от деления при выполнении инструкции `"div ebx"`? • Остаток от деления записывается в регистр `edx`. 6. Для чего используется инструкция `"inc edx"`? • Инструкция `"inc edx"` используется для увеличения значения в регистре `edx` на `1`, в соответствии с формулой вычисления варианта. 7. Какие строки листинга отвечают за вывод на экран результата вычислений? • Строка `"mov eax, edx"` перекладывает результат вычислений в регистр `eax`. Строка `"call iprintLF"` вызывает подпрограмму для вывода значения на экран.

## **4 Выводы**

В ходе данной лабораторной работы я изучила работу с арифметическими операциями.