

# **Отчет по лабораторной работе №7**

**Дисциплина: архитектура компьютера**

Игнатенкова В. Н

# Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Выполнение заданий для самостоятельной работы	15
4	Выводы	18

# Список иллюстраций

2.1	Создание файла . . . . .	6
2.2	Запись программы в файл . . . . .	6
2.3	Копирование файла . . . . .	7
2.4	Запуск исполняемого файла . . . . .	7
2.5	Изменение программы . . . . .	7
2.6	Запуск исполняемого файла . . . . .	8
2.7	Создание файла . . . . .	8
2.8	Запись программы в файл . . . . .	8
2.9	Запуск исполняемого файла . . . . .	9
2.10	Изменение программы . . . . .	9
2.11	Запуск исполняемого файла . . . . .	9
2.12	Изменение программы . . . . .	10
2.13	Создание программы . . . . .	10
2.14	Создание программы . . . . .	11
2.15	Запуск исполняемого файла . . . . .	11
2.16	Изменение программы . . . . .	12
2.17	Запуск исполняемого файла . . . . .	12
2.18	Создание программы . . . . .	13
2.19	Запуск исполняемого файла . . . . .	13
3.1	Создание программы . . . . .	15
3.2	Запуск исполняемого файла . . . . .	16
3.3	Проверка работы программы . . . . .	16

## **Список таблиц**

# 1 Цель работы

Цель данной лабораторной работы - освоение арифметических инструкций языка ассемблера NASM

## 2 Выполнение лабораторной работы

Создала директорию lab07, переместилась в нее, создала файл lab7-1.asm, проверила, что файл был создан (рис. 2.1)

```
(vnignatenkova@vnignatenkova)-[~]  
$ cd work/study/2022-2023/Архитектура\ Компьютера/study_2022-2023_arh-pc/  
(vnignatenkova@vnignatenkova)-[~/work/study/2022-2023/Архитектура Компьютера/study_2022-2023_arh-pc]  
$ mkdir lab07  
(vnignatenkova@vnignatenkova)-[~/work/study/2022-2023/Архитектура Компьютера/study_2022-2023_arh-pc/lab07]  
$ cd lab07  
(vnignatenkova@vnignatenkova)-[~/work/study/2022-2023/Архитектура Компьютера/study_2022-2023_arh-pc/lab07]  
$ touch lab7-1.asm
```

Рис. 2.1: Создание файла

Записала в файл lab7-1.asm программу (рис. 2.2).

```
~/work/study/2022-2023/Архитектура Компьютера/study_2022-2023_arh-pc/lab07/lab7-1.asm  
Файл  Правка  Поиск  Вид  Документ  Справка  
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 sprintf  
13 call quit  
14 |
```

Рис. 2.2: Запись программы в файл

Копирую в текущий каталог файл in\_out.asm (рис. 2.3).

```
(vnignatenkova@vnignatenkova)-[~/work/study/2022-2023/Архитектура Компьютера/lab07/lab7-1]
$ cp ~/Загрузки/in_out.asm in_out.asm
```

Рис. 2.3: Копирование файла

Создаю исполняемый файл программы и запускаю его (рис. 2.4). Программа вывела символ j, потому что он соответствует сумме двоичных кодов символов 4 и 6 по системе ASCII.

```
(vnignatenkova@vnignatenkova)-[~/work/study/2022-2023/Архитектура Компьютера/lab07/lab7-1]
$ nasm -f elf lab7-1.asm

(vnignatenkova@vnignatenkova)-[~/work/study/2022-2023/Архитектура Компьютера/lab07/lab7-1]
$ ld -m elf_i386 -o lab7-1 lab7-1.o

(vnignatenkova@vnignatenkova)-[~/work/study/2022-2023/Архитектура Компьютера/lab07/lab7-1]
$ ./lab7-1
j
```

Рис. 2.4: Запуск исполняемого файла

Изменяю программу (рис. 2.5).

```
~/work/study/2022-2023/Архитектура Компьютера/study_2022-2023_arh-pc/lab07/lab7-1.asm
Файл  Правка  Поиск  Вид  Документ  Справка
[Icons]
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 sprintf
13 call quit
14
```

Рис. 2.5: Изменение программы

Создаю исполняемый файл программы и запускаю его (рис. 2.6). Программа вывела символ перевода строки, потому что он соответствует коду 10=6+4 по системе ASCII.

```

(vnignatenkova@vnignatenkova)-[~/work/study/2022-2023/Архитек
$ nasm -f elf lab7-1.asm

(vnignatenkova@vnignatenkova)-[~/work/study/2022-2023/Архитек
$ ld -m elf_i386 -o lab7-1 lab7-1.o

(vnignatenkova@vnignatenkova)-[~/work/study/2022-2023/Архитек
$ ./lab7-1

```

Рис. 2.6: Запуск исполняемого файла

Создаю файл lab7-2.asm (рис. 2.7).

```

(vnignatenkova@vnignatenkova)-[~/work/study/2022-2023/
$ touch lab7-2.asm

```

Рис. 2.7: Создание файла

Ввожу текст программы в созданный файл (рис. 2.8).

```

~/work/study/2022-2023/Архитектура Компьютера/study_2022-2023_arh-pc/lab07/lab7-2.asm
Файл  Правка  Поиск  Вид  Документ  Справка
+  [иконки]  [иконки]  [иконки]  [иконки]  [иконки]  [иконки]  [иконки]  [иконки]  [иконки]
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. 2.8: Запись программы в файл

Создаю и запускаю исполняемый файл lab7-2 (рис. 2.9). Теперь программа выводит число, а не символ, но это число - сложение кодов символов “6” и “4”.



Рис. 2.9: Запуск исполняемого файла

Изменяю программу (рис. 2.10).

Рис. 2.10: Изменение программы

Запускаю новый исполняемый файл, теперь складываются именно цифры (рис. 11).

Рис. 2.11: Запуск исполняемого файла

Заменяю в тексте программы функцию `iprintLF` на `iprint` (рис. 2.12).

```

(vnignatenkova@vnignatenkova)-[~/work/study/2022-2023/Архитектура Ко
$ ./lab7-4
Введите значение переменной x: 1
Результат: 13
(vnignatenkova@vnignatenkova)-[~/work/study/2022-2023/Архитектура Ко
$ ./lab7-4
Введите значение переменной x: 5
Результат: 25

```

Рис. 2.12: Изменение программы

Создаю и запускаю новый исполняемый файл (рис. 2.13). Вывод не изменился, потому что символ переноса строки не отображался, когда программа исполнялась с функцией `iprintLF`, а `iprint` не добавляет к выводу символ переноса строки.

```

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 iprint
9 call quit
10

```

Рис. 2.13: Создание программы

Создаю файл `lab7-3.asm` и в нем записываю программу (рис. 2.14).

```
(vnignatenkova@vnignatenkova)-[~/work/study/2022-2023/Архитектура Компьютера/s
$ touch lab7-3.asm

~/work/study/2022-2023/Архитектура Компьютера/study_2022-2023_arh-pc/lab07/lab7-3.asm
Файл Правка Поиск Вид Документ Справка

1 %include 'in_out.asm' ; подключение внешнего файла
2 SECTION .data
3 div: DB 'Результат: ',0
4 rem: DB 'Остаток от деления: ',0
5 SECTION .text
6 GLOBAL _start
7 _start:
8 ; ————— Вычисление выражения
9 mov eax,5 ; EAX=5
10 mov ebx,2 ; EBX=2
11 mul ebx ; EAX=EAX*EBX
12 add eax,3 ; EAX=EAX+3
13 xor edx,edx ; обнуляем EDX для корректной работы div
14 mov ebx,3 ; EBX=3
15 div ebx ; EAX=EAX/3, EDX=остаток от деления
16 mov edi,eax ; запись результата вычисления в 'edi'
17 ; ————— Вывод результата на экран
18 mov eax,div ; вызов подпрограммы печати
19 call sprint ; сообщения 'Результат: '
20 mov eax,edi ; вызов подпрограммы печати значения
21 call iprintLF ; из 'edi' в виде символов
22 mov eax,rem ; вызов подпрограммы печати
23 call sprint ; сообщения 'Остаток от деления: '
24 mov eax,edx ; вызов подпрограммы печати значения
25 call iprintLF ; из 'edx' (остаток) в виде символов
26 call quit ; вызов подпрограммы завершения
27
```

Рис. 2.14: Создание программы

Создаю исполняемый файл и запускаю его (рис. 2.15).

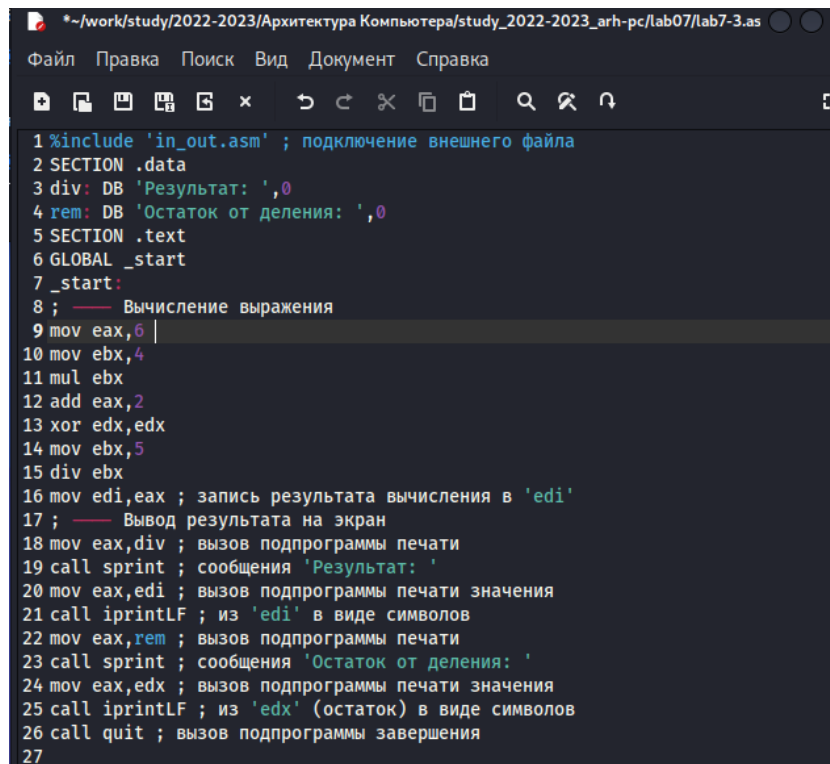
```
(vnignatenkova@vnignatenkova)-[~/work/st
$ nasm -f elf lab7-3.asm

(vnignatenkova@vnignatenkova)-[~/work/st
$ ld -m elf_i386 -o lab7-3 lab7-3.o

(vnignatenkova@vnignatenkova)-[~/work/st
$ ./lab7-3
Результат: 4
Остаток от деления: 1
```

Рис. 2.15: Запуск исполняемого файла

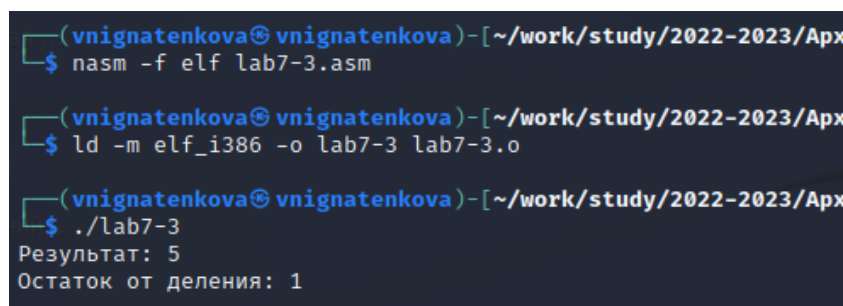
Изменяю текст программы, чтобы посчитать значение выражения  $(4*6+2)/5$  (рис. 2.16).



```
*~/work/study/2022-2023/Архитектура Компьютера/study_2022-2023_arh-pc/lab07/lab7-3.as
Файл Правка Поиск Вид Документ Справка
1 %include 'in_out.asm' ; подключение внешнего файла
2 SECTION .data
3 div: DB 'Результат: ',0
4 rem: DB 'Остаток от деления: ',0
5 SECTION .text
6 GLOBAL _start
7 _start:
8 ; ——— Вычисление выражения
9 mov eax,6
10 mov ebx,4
11 mul ebx
12 add eax,2
13 xor edx,edx
14 mov ebx,5
15 div ebx
16 mov edi,eax ; запись результата вычисления в 'edi'
17 ; ——— Вывод результата на экран
18 mov eax,div ; вызов подпрограммы печати
19 call sprint ; сообщения 'Результат: '
20 mov eax,edi ; вызов подпрограммы печати значения
21 call iprintLF ; из 'edi' в виде символов
22 mov eax,rem ; вызов подпрограммы печати
23 call sprint ; сообщения 'Остаток от деления: '
24 mov eax,edx ; вызов подпрограммы печати значения
25 call iprintLF ; из 'edx' (остаток) в виде символов
26 call quit ; вызов подпрограммы завершения
27
```

Рис. 2.16: Изменение программы

Создаю исполняемый файл, запускаю его, программа работает корректно (рис. 2.17).



```
(vsnignatenkova@vsnignatenkova)-[~/work/study/2022-2023/Арх
$ nasm -f elf lab7-3.asm

(vsnignatenkova@vsnignatenkova)-[~/work/study/2022-2023/Арх
$ ld -m elf_i386 -o lab7-3 lab7-3.o

(vsnignatenkova@vsnignatenkova)-[~/work/study/2022-2023/Арх
$ ./lab7-3
Результат: 5
Остаток от деления: 1
```

Рис. 2.17: Запуск исполняемого файла

Создаю файл variant.asm и записываю в него текст программы для вычисления варианта (рис. 2.18).

```
(vnignatenkova@vnignatenkova)-[~/work/study/2022-2023/Архитектура Компьютера]
$ touch lab7-4.asm

~/work/study/2022-2023/Архитектура Компьютера/study_2022-2023_arh-pc/lab07/lab7-4.asm
Файл Правка Поиск Вид Документ Справка

1 %include 'in_out.asm' ; подключение внешнего файла
2 SECTION .data ; секция инициализированных данных
3 msg: DB 'Введите значение переменной x: ',0
4 rem: DB 'Результат: ',0
5 SECTION .bss ; секция не инициализированных данных
6 x: RESB 80 ; Переменная, значение к-рой будем вводить с клавиатуры,
   выделенный размер - 80 байт
7 SECTION .text ; Код программы
8 GLOBAL _start ; Начало программы
9 _start: ; Точка входа в программу
10 ; ————— Вычисление выражения
11 mov eax, msg ; запись адреса выводимого сообщения в eax
12 call sprint ; вызов подпрограммы печати сообщения
13 mov ecx, x ; запись адреса переменной в ecx
14 mov edx, 80 ; запись длины вводимого значения в edx
15 call sread ; вызов подпрограммы ввода сообщения
16 mov eax, x ; вызов подпрограммы преобразования
17 call atoi ; ASCII кода в число, `eax=x`
18 add eax, 10;
19 mov ebx, 3 ;
20 mul ebx; |
21 add eax, -20;
22 mov edi, eax ; запись результата вычисления в `edi`
23 ; ————— Вывод результата на экран
24 mov eax, rem ; вызов подпрограммы печати
25 call sprint ; сообщения 'Результат: '
26 mov eax, edi ; вызов подпрограммы печати значения
27 call sprint ; на `edi` в виде числа
```

Рис. 2.18: Создание программы

Создаю исполняемый файл и запускаю его. Мой номер варианта для выполнения заданий для самостоятельной работы - 18 (рис. 2.19).

```
(vnignatenkova@vnignatenkova)-[~/work/study/2022-2023/Архитектура
$ nasm -f elf variant.asm

(vnignatenkova@vnignatenkova)-[~/work/study/2022-2023/Архитектура
$ ld -m elf_i386 -o variant variant.o

(vnignatenkova@vnignatenkova)-[~/work/study/2022-2023/Архитектура
$ ./variant
Введите No студенческого билета:
1132226497
Ваш вариант: 18
```

Рис. 2.19: Запуск исполняемого файла

1. За вывод сообщения “Ваш вариант” отвечают строки кода:

```
mov eax, rem
call sprint
```

2. Инструкция `mov ecx, x` используется, чтобы положить адрес вводимой строки `x` в регистр `ecx` `mov edx, 80` - запись в регистр `edx` длины вводимой строки `call sread` - вызов подпрограммы из внешнего файла, обеспечивающей ввод сообщения с клавиатуры
3. `call atoi` используется для вызова подпрограммы из внешнего файла, которая преобразует `ascii`-код символа в целое число и записывает результат в регистр `eax`
4. За вычисления варианта отвечают строки:

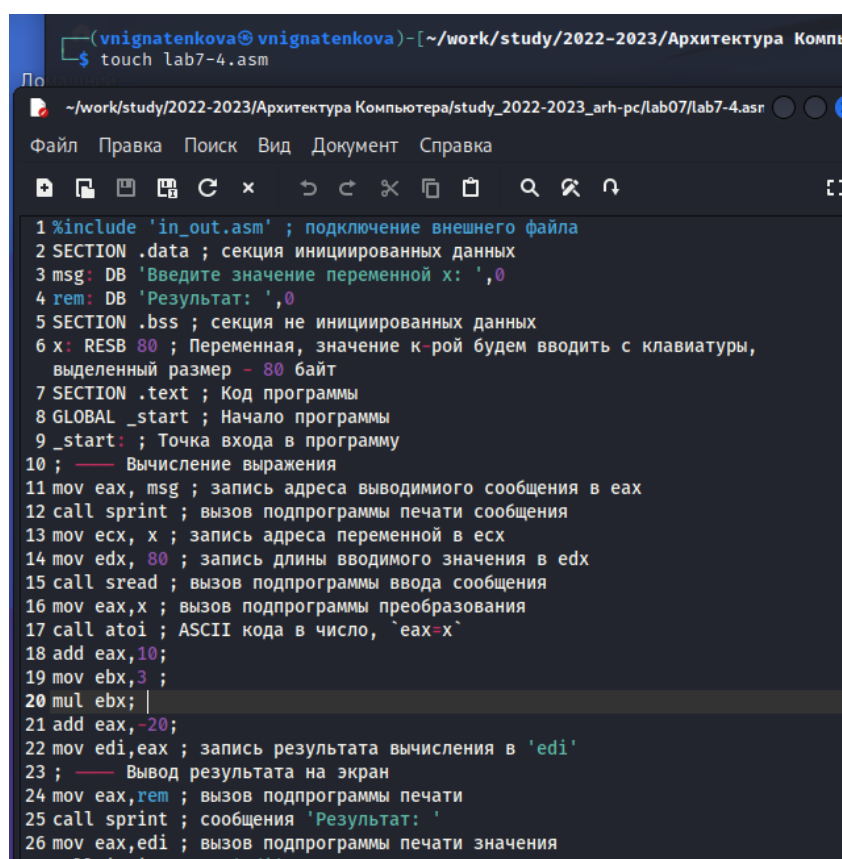
```
xor  edx,edx ; обнуление edx для корректной работы div
mov  ebx,20 ; ebx = 20
div  ebx ; eax = eax/20, edx - остаток от деления
inc  edx ; edx = edx + 1
```

5. При выполнении инструкции `div ebx` остаток от деления записывается в регистр `edx`
6. Инструкция `inc edx` увеличивает значение регистра `edx` на 1
7. За вывод на экран результатов вычислений отвечают строки:

```
mov  eax,edx
call iprintLF
```

### 3 Выполнение заданий для самостоятельной работы

Создала файл lab7-4.asm, в нем написала программу для вычисления выражения из варианта 18:  $3 \cdot (x + 10) - 20$  (рис. 3.1).



```
1 %include 'in_out.asm' ; подключение внешнего файла
2 SECTION .data ; секция иницированных данных
3 msg: DB 'Введите значение переменной x: ',0
4 rem: DB 'Результат: ',0
5 SECTION .bss ; секция не иницированных данных
6 x: RESB 80 ; Переменная, значение которой будем вводить с клавиатуры,
   ; выделенный размер - 80 байт
7 SECTION .text ; Код программы
8 GLOBAL _start ; Начало программы
9 _start: ; Точка входа в программу
10 ; ————— Вычисление выражения
11 mov eax, msg ; запись адреса выводимого сообщения в eax
12 call sprint ; вызов подпрограммы печати сообщения
13 mov ecx, x ; запись адреса переменной в ecx
14 mov edx, 80 ; запись длины вводимого значения в edx
15 call sread ; вызов подпрограммы ввода сообщения
16 mov eax, x ; вызов подпрограммы преобразования
17 call atoi ; ASCII кода в число, 'eax=x'
18 add eax, 10;
19 mov ebx, 3 ;
20 mul ebx; |
21 add eax, -20;
22 mov edi, eax ; запись результата вычисления в 'edi'
23 ; ————— Вывод результата на экран
24 mov eax, rem ; вызов подпрограммы печати
25 call sprint ; сообщения 'Результат: '
26 mov eax, edi ; вызов подпрограммы печати значения
27 call iexit ; на 'edi' с указателем
```

Рис. 3.1: Создание программы

Создала исполняемый файл и запустила его. Программа работает коррект-

но(рис. 3.2).

```
(vnignatenkova@vnignatenkova)-[~/work/study/2022-2023/Архитектура Ко  
$ nasm -f elf lab7-4.asm  
  
(vnignatenkova@vnignatenkova)-[~/work/study/2022-2023/Архитектура Ко  
$ ld -m elf_i386 -o lab7-4 lab7-4.o  
  
(vnignatenkova@vnignatenkova)-[~/work/study/2022-2023/Архитектура Ко  
$ ./lab7-4  
Введите значение переменной x: 10  
Результат: 40
```

Рис. 3.2: Запуск исполняемого файла

Проверила работу программы для значений x1 и x2 (рис. 3.3).

```
(vnignatenkova@vnignatenkova)-[~/work/study/2022-2023/Архитектура Ко  
$ ./lab7-4  
Введите значение переменной x: 1  
Результат: 13  
  
(vnignatenkova@vnignatenkova)-[~/work/study/2022-2023/Архитектура Ко  
$ ./lab7-4  
Введите значение переменной x: 5  
Результат: 25
```

Рис. 3.3: Проверка работы программы

### Код написанной программы:

```
%include 'in_out.asm' ; подключение внешнего файла  
SECTION .data ; секция инициированных данных  
msg: DB 'Введите значение переменной x: ',0  
rem: DB 'Результат: ',0  
SECTION .bss ; секция не инициированных данных  
x: RESB 80 ; Переменная, значение к-рой будем вводить с клавиатуры, выделенный ра  
SECTION .text ; Код программы  
GLOBAL _start ; Начало программы  
_start: ; Точка входа в программу  
; ---- Вычисление выражения  
mov eax, msg ; запись адреса выводимого сообщения в eax  
call sprint ; вызов подпрограммы печати сообщения
```



```
mov ecx, x ; запись адреса переменной в ecx
mov edx, 80 ; запись длины вводимого значения в edx
call sread ; вызов подпрограммы ввода сообщения
mov eax, x ; вызов подпрограммы преобразования
call atoi ; ASCII кода в число, `eax=x`
add eax, 10;
mov ebx, 3 ;
mul ebx;
add eax, -20;
mov edi, eax ; запись результата вычисления в 'edi'
; ---- Вывод результата на экран
mov eax, rem ; вызов подпрограммы печати
call sprint ; сообщения 'Результат: '
mov eax, edi ; вызов подпрограммы печати значения
call iprint ; из 'edi' в виде символов
call quit ; вызов подпрограммы завершения
```

## 4 Выводы

В ходе выполнения лабораторной работы я освоила арифметические инструкции языка ассемблера NASM.