

Отчет по лабораторной работе №6

Дисциплина: Архитектура компьютера

-04-24

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
3.1	Символьные и численные данные в NASM	7
3.2	Выполнение арифметических операций в NASM	10
3.3	Ответы на вопросы	13
3.4	Задание для самостоятельной работы	14
4	Выводы	16

Список иллюстраций

3.1	Создание файла для лабораторной работы	7
3.2	Результат	7
3.3	Текст программы	8
3.4	Таблица ASCII	9
3.5	Создаю файл	9
3.6	Результат lab6-2	10
3.7	Результат	10
3.8	Результат	10
3.9	Текст программы	11
3.10	Результат	11
3.11	Текст программы	12
3.12	Результат	12
3.13	Текст программы	13
3.14	Вычленение результата	13
3.15	Текст программы	15
3.16	Результат	15

Список таблиц

1 Цель работы

Освоение арифметических инструкций языка ассемблера NASM.

2 Задание

1. Символьные и численные данные в NASM
2. Выполнение арифметических операций в NASM
3. Ответы на вопросы
4. Задание для самостоятельной работы

3 Выполнение лабораторной работы

3.1 Символьные и численные данные в NASM

Создаю каталог для программ лабораторной работы № 6, затем перехожу в него и создайте файл lab6-1.asm: (рис. 3.1).

```
esandan@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs$ cd lab06
esandan@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab06$ cd report
esandan@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab06/report$ mkdir lab06
esandan@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab06/report$ cd lab06
esandan@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab06/report/lab06$ touch lab6-1.asm
esandan@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab06/report/lab06$ mc
```

Рис. 3.1: Создание файла для лабораторной работы

Ввожу в этот файл текст из листинга 6.1 и вывожу результат. Создаю исполняемый файл и запускаю его, вот какой результат у меня получился (рис. 3.2).

```
esandan@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab06/report/lab06$ nasm -f elf lab6-1.asm
esandan@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab06/report/lab06$ ld -m elf_i386 -o lab6-1 lab6-1.o
esandan@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab06/report/lab06$ ./lab6-1
j
esandan@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab06/report/lab06$
```

Рис. 3.2: Результат

В результате вывело символ j. Несмотря на то, что мы ожидали увидеть число 10

Далее я изменяю текст программы и вместо символов записываю в регистры числа. (рис. 3.3).

```

Lab6-1.asm [-M--
%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

```

Рис. 3.3: Текст программы

В результате получается снова не число, вывелся символ с кодом 10. Используя таблицу ASCII я определила, что на выводе мы получили пустой символ, которому как раз таки соответствует число 10 в таблице(рис. 3.4).

ASCII Table

Dec	Hex	Oct	Char	Dec
0	0	0		32
1	1	1		33
2	2	2		34
3	3	3		35
4	4	4		36
5	5	5		37
6	6	6		38
7	7	7		39
8	8	10		40
9	9	11		41
10	A	12		42
11	B	13		43
12	C	14		44
13	D	15		45
14	E	16		46
15	F	17		47
16	10	20		48
17	11	21		49
18	12	22		50

Рис. 3.4: Таблица ASCII

Теперь создаю файл lab6-2.asm (рис. 3.5).

```

hesand@fedora: /work/study/2024-2025/Архитектура_компьютера/arch-pc/labs/lab06/report/lab06$ nano -f elf lab6-2.o
hesand@fedora: /work/study/2024-2025/Архитектура_компьютера/arch-pc/labs/lab06/report/lab06$ ld -m elf_i386 -o la
b6-2 lab6-2.o
hesand@fedora: /work/study/2024-2025/Архитектура_компьютера/arch-pc/labs/lab06/report/lab06$ ./lab6-2
106
hesand@fedora: /work/study/2024-2025/Архитектура_компьютера/arch-pc/labs/lab06/report/lab06$

```

Рис. 3.5: Создаю файл

Ввожу туда текст с листинга 6.2. Затем запускаю программу и получаю резуль-
тат в виде числа 106(рис. 3.6).

```

aesandan@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab06/report/lab06$ nasm -f elf lab6-2.asm
aesandan@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab06/report/lab06$ ld -m elf_i386 -o lab6-2 lab6-2.o
aesandan@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab06/report/lab06$ ./lab6-2
106
aesandan@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab06/report/lab06$

```

Рис. 3.6: Результат lab6-2

По аналогии с прошлым примером я изменяю текст программы, заменяя символы на числа, и в таком случае в результате я получу число 10 (рис. 3.7).

```

aesandan@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab06/report/lab06$ nasm -f elf lab6-2.asm
aesandan@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab06/report/lab06$ ld -m elf_i386 -o lab6-2 lab6-2.o
aesandan@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab06/report/lab06$ ./lab6-2
10
aesandan@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab06/report/lab06$

```

Рис. 3.7: Результат

А если я изменю функцию `iprintLF` на `iprint`, то тогда командная строка будет на той же строке, что и вывод (рис. 3.8).

```

aesandan@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab06/report/lab06$ nasm -f elf lab6-2.asm
aesandan@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab06/report/lab06$ ld -m elf_i386 -o lab6-2 lab6-2.o
aesandan@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab06/report/lab06$ ./lab6-2
10
aesandan@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab06/report/lab06$

```

Рис. 3.8: Результат

3.2 Выполнение арифметических операций в NASM

Я создаю новый файл `lab6-3.asm` для работы с листингом 6.3, который я соответственно ввожу в этот файл (рис. 3.9).

```

lab6-3.asm      [----] 41 L:[ 10+16  26/ 26] *(1236/1236b) <EOF>
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,edx ; вызов подпрограммы печати значения
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

```

Рис. 3.9: Текст программы

Заметим, что вот такой результат выдает нам программа (рис. 3.10).

```

sm
aesandan@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab06/report/lab06$ ld -m elf_i386 -o lab6-3 lab6-3.o
aesandan@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab06/report/lab06$ ./lab6-3
Результат: 4
Остаток от деления: 1
aesandan@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab06/report/lab06$

```

Рис. 3.10: Результат

В соответствии с заданием я изменяю текст программы для вычисления выражения $f(x)=(4*6+2)/5$ (рис. 3.11).

```

#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,4 ; EAX=4
mov ebx,6 ; EBX=6
mul ebx ; EAX=EAX*EBX
add eax,2 ; EAX=EAX+2
xor edx,edx ; обнуляем EDX для корректной работы div
mov ebx,5 ; EBX=5
div ebx ; EAX=EAX/5, 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

```

Рис. 3.11: Текст программы

Проверяю его работу. (рис. 3.12).

```

sm
aesandam@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab06/report/lab06$ ld -m elf_i386 -o lab6-3 lab6-3.o
aesandam@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab06/report/lab06$ ./lab6-3
Результат: 5
Остаток от деления: 1
aesandam@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab06/report/lab06$

```

Рис. 3.12: Результат

Для выполнения следующего этапа лабораторной работы, я создаю файл variant.asm и ввожу текст из листинга 6.4, который вычисляет вариант задания по номеру студенческого билета (рис. 3.13).

```

#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 ; ASCII кода в число, 'eax=x'
Архитектура ЭВМ
xor edx, edx
mov ebx, 20
div ebx
inc edx
mov eax, rem
call sprint
mov eax, edx
call iprintLF
call quit

```

Рис. 3.13: Текст программы

Вычисляю свой номер варианта (рис. 3.14).

```

Ваш вариант: 2
mesand@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab06/report/lab06$

```

Рис. 3.14: Вычисление результата

3.3 Ответы на вопросы

1. В листинге 6.4 за вывод сообщения отвечают строки

```
mov eax, rem
```

```
call sprint
```

2. первая инструкция `mov esx, x` используется, чтобы положить адрес вводимой строки `x` в регистр `esx`. а строка `mov edx, 80` - записывает в регистр `edx` длины вводимой строки. ну и строка `call sread` - вызывает подпрограмму из внешнего файла, обеспечивающей ввод сообщения с клавиатуры
3. инструкция `call atoi` используется в случае, когда надо преобразовать символы в целые числа
4. За вычисления варианта в листинге 6.4 отвечают следующие 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`” используется для увеличения значения регистра на 1
7. За вывод на экран результата отвечают эти строки:

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

3.4 Задание для самостоятельной работы

Мне необходимо написать программу, которая выводит на экран выражения для вычисления, а также выводит результат вычислений, при этом программа сама должна посчитать заданное выражение в зависимости от введенных мной

переменных. Я буду выполнять задание в соответствии с вариантом №2 , который я вычислила в ходе лабораторной работы. (рис. 3.15).

```
nomer.asm      [----]  9 L:[ 1+27  28/ 29] *(538 / 539b) 0010 0x00A
#include 'in_out.asm'
SECTION .data
msg: DB 'Выражение для вычисления:(12x+3)5. Введите значение x: ',0
rem: DB 'Ответ: ',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 ; вызов подпрограммы преобразования
call atoi ; ASCII кода в число, 'eax=x'

mov ebx,12.
mul ebx
add eax, 3
mov ecx, 5
mul ecx
mov edi,eax..
mov eax,rem
call sprint
mov eax,edi
call iprintLF
call quit
```

Рис. 3.15: Текст программы

После создания исполняемого файла, я ввожу переменные такие, как в таблице 6.3 и проверяю как работает программа (рис. 3.16).

```
aesandan@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab06/report/lab06$ ./lab6
Выражение для вычисления:(12x+3)5. Введите значение x:
1
Ответ: 75
aesandan@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab06/report/lab06$ nasm -f elf lab6.asm
aesandan@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab06/report/lab06$ ld -m elf_i386 -o lab6 lab6.o
aesandan@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab06/report/lab06$ ./lab6
Выражение для вычисления:(12x+3)5. Введите значение x:
3
Ответ: 195
aesandan@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab06/report/lab06$
```

Рис. 3.16: Результат

Программа работает правильно!

4 Выводы

Я освоила арифметические инструкции языка ассемблера NASM.