Отчет по лабораторной работе №6

Дисциплина: Архитектура компьютера

Максимова Дарья Валерьевна

Содержание

Список иллюстраций

Список таблиц

1 Цель работы

Освоение арифмитических инстуркций 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 программам лабораторной работы № 6, затем перехожу в него и создайте файл lab6-1.asm: (рис. 3.1).

```
dvmaksimova@dk3n55 ~ $ mkdir ~/work/arch-pc/lab06
dvmaksimova@dk3n55 ~ $ ~/work/arch-pc/lab06
bash: /afs/.dk.sci.pfu.edu.ru/home/d/v/dvmaksimova/work/arch-pc/lab06: Э
or
dvmaksimova@dk3n55 ~ $ cd ~/work/arch-pc/lab06
dvmaksimova@dk3n55 ~/work/arch-pc/lab06 $ touch lab6-1.asm
dvmaksimova@dk3n55 ~/work/arch-pc/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).

```
dvmaksimova@dk3n60 ~/work/arch-pc/lab06 $ ./lab6-1
j
```

Рис. 3.2: Результат

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

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

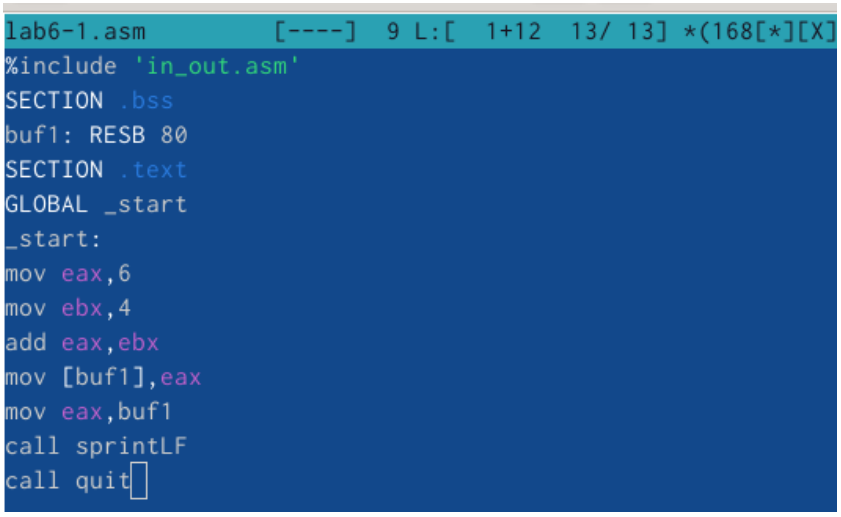


Рис. 3.3: Текст программы

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 соответсувуте 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

Рис. 3.4: Таблица ASCII

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

```
dvmaksimova@dk3n60 ~/work/arch-pc/lab06 $ touch lab6-2.asm
dvmaksimova@dk3n60 ~/work/arch-pc/lab06 $
```

Рис. 3.5: Создаю файл

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

```
dvmaksimova@dk3n60 ~/work/arch-pc/lab06 $ nasm -f elf lab6-2.asm
dvmaksimova@dk3n60 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab602 lab6-2.o
dvmaksimova@dk3n60 ~/work/arch-pc/lab06 $ ./lab6-2
bash: ./lab6-2: Нет такого файла или каталога
dvmaksimova@dk3n60 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-2 lab6-2.o
dvmaksimova@dk3n60 ~/work/arch-pc/lab06 $ ./lab6-2
106
dvmaksimova@dk3n60 ~/work/arch-pc/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 резутате я получу число 10 (рис. 3.7).

```
dvmaksimova@dk3n60 ~/work/arch-pc/lab06 $ mc

dvmaksimova@dk3n60 ~/work/arch-pc/lab06 $ nasm -f elf lab6-2.asm
dvmaksimova@dk3n60 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-2 lab6-2.o
dvmaksimova@dk3n60 ~/work/arch-pc/lab06 $ ./lab6-2
10
```

Рис. 3.7: Результат

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

```
dvmaksimova@dk3n60 ~/work/arch-pc/lab06 $ mc

dvmaksimova@dk3n60 ~/work/arch-pc/lab06 $ nasm -f elf lab6-2.asm
dvmaksimova@dk3n60 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-2 lab6-2.o
dvmaksimova@dk3n60 ~/work/arch-pc/lab06 $ ./lab6-2
10dvmaksimova@dk3n60 ~/work/arch-pc/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,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. 3.9: Текст программы

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

```
dvmaksimova@dk3n60 ~/work/arch-pc/lab06 $ nasm -f elf lab6-3.asm
dvmaksimova@dk3n60 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-3 lab6-3.o
dvmaksimova@dk3n60 ~/work/arch-pc/lab06 $ ./lab6-3
Результат: 4
Остаток от деления: 1
dvmaksimova@dk3n60 ~/work/arch-pc/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).

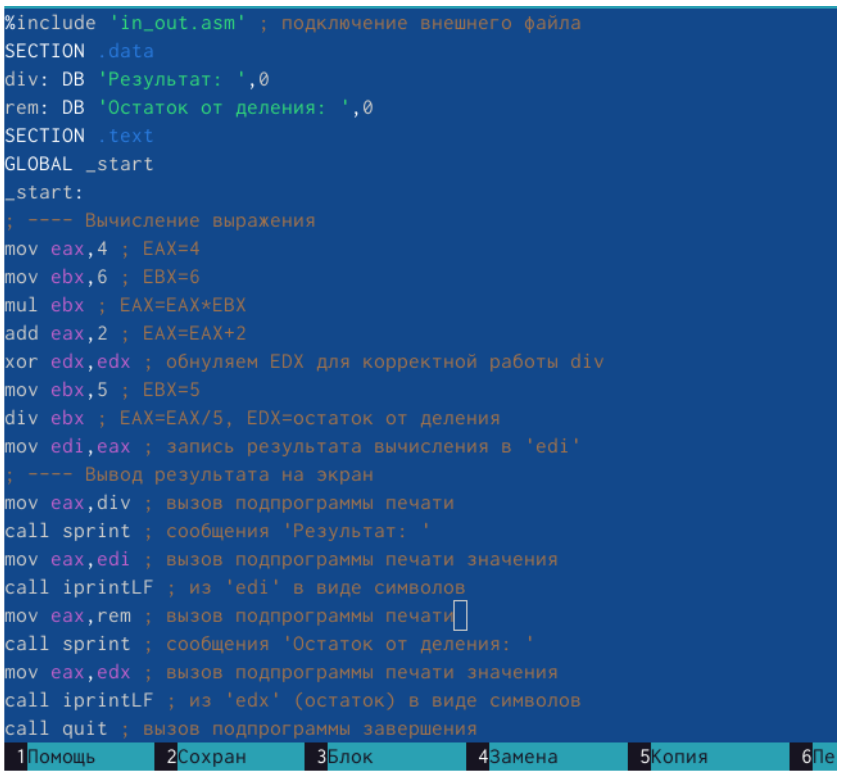


Рис. 3.11: Текст программы

Проверяю его работу. (рис. 3.12).

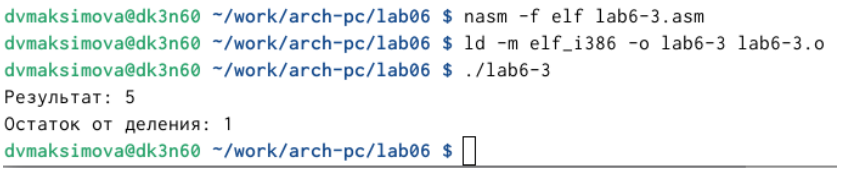


Рис. 3.12: Результат

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

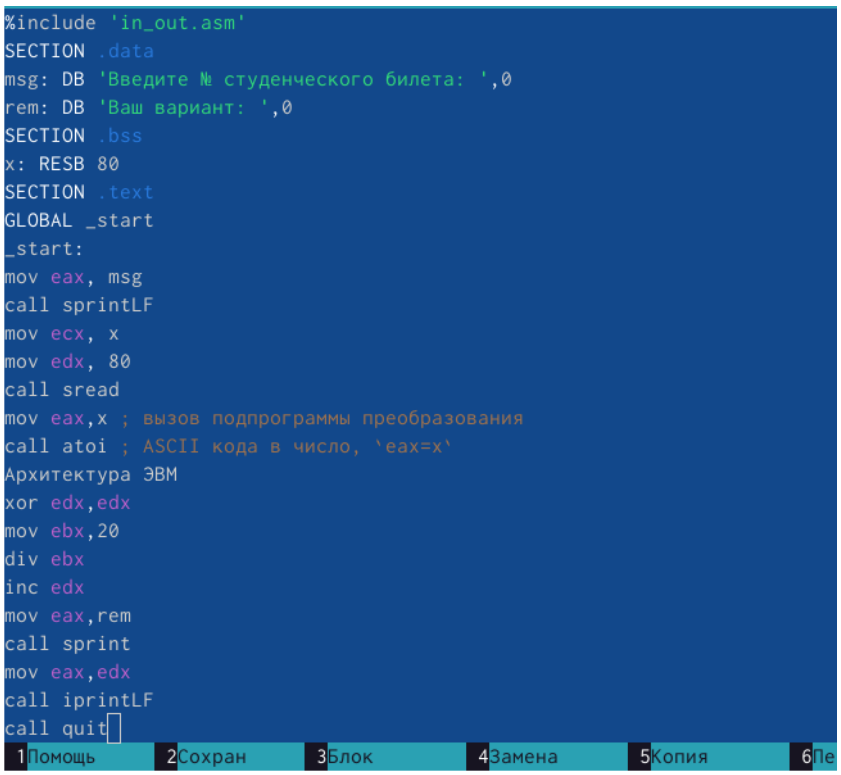


Рис. 3.13: Текст программы

Вычисляю свой номер варианта (рис. 3.14).

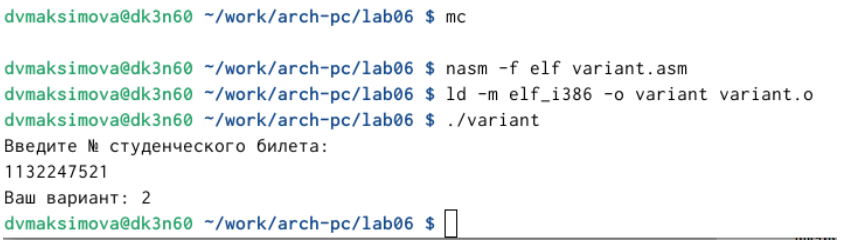


Рис. 3.14: Вычиление результата

3.3 Ответы на вопросы

1. В листинге 6.4 за вывод сообщения отвечают строчки

mov eax,rem

call sprint

2. первая инструкция mov ecx, x используется, чтобы положить адрес вводимой строки x в регистр . а строчка ecx mov edx, 80 - записывает в регистр edx длины вводимой строки . ну и строчка 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 строчки:

```
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

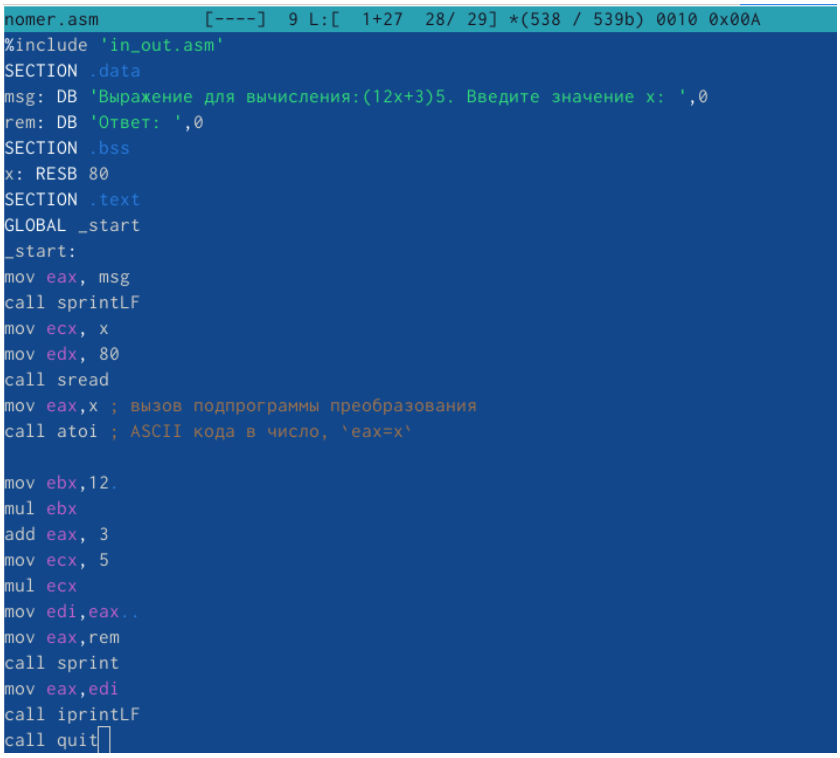
7. За вывод на экран результата отвечают эти строчки:

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

3.4 Задание для самостоятельной работы

Мне необходимо написать программу, которая выводит на экран выражения для вычисления, а также выводить 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).

```
dvmaksimova@dk3n60 ~/work/arch-pc/lab06 $ nasm -f elf nomer.asm
dvmaksimova@dk3n60 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o nomer  nomer.o
dvmaksimova@dk3n60 ~/work/arch-pc/lab06 $ ./nomer
Выражение для вычисления:(12x+3)5. Введите значение x:
1
Ответ: 75
dvmaksimova@dk3n60 ~/work/arch-pc/lab06 $ ./nomer
Выражение для вычисления:(12x+3)5. Введите значение x:
6
Ответ: 375
dvmaksimova@dk3n60 ~/work/arch-pc/lab06 $
```

Рис. 3.16: Результат

Программа работает успешно!

4 Выводы

Я освоила арифмитические инстуркции языка ассемблера NASM.