

Лабораторная работа №7

Щетинин Даниил Николаевич

Содержание

1	Цель работы	4
2	Задание	5
3	Выполнение лабораторной работы	6
4	Ответы на вопросы 1-7	8
5	Задание для самостоятельной работы	10
6	Выводы	15

Список иллюстраций

5.1	lab7-1	11
5.2	lab7-1	11
5.3	lab7-2	11
5.4	lab7-2	11
5.5	lab7-2	11
5.6	работа файла lab7-3	12
5.7	Код новой lab7-3.asm	12
5.8	Результат новой lab7-3	13
5.9	Полученный вариант	13
5.10	код var16	14

1 Цель работы

Освоение арифметических инструкций языка ассемблера NASM.

2 Задание

Написать программу вычисления функции, ознакомиться с Арифметическими операциями используя язык ассемблера NASM

3 Выполнение лабораторной работы

Шаг 1

Создаём каталог для программ лабораторной работы No 7, и в нём создаём файл lab7-1.asm и переносим туда файл in_out.asm, копируем в него текст листинга 7.1, для того, чтобы ознакомиться с принципом вывода переменных на экран. Создаём исполняемый файл и получаем результат, символ 'j', т.к. код полученного нами элемента является кодом не для числа 10, а для символа 'j'

(рис. 5.1)

Шаг 2

Изменим текст нашего файла, заменив

```
mov eax, '6'
```

```
mov ebx, '4'
```

на

```
mov eax, 6
```

```
mov ebx, 4
```

Получим 10 символ ASCII, который соответствует символу 'LF,/n', который не отображается на экране.

(рис. 5.2)

Шаг 3

Создадим следующий файл, lab7-2.asm, где мы используем код из файла in_out.asm для преобразования ASCII символов в числа. Копируем код из листинга 7.2, запустим исполняемый файл, :

(рис. 5.3)

Получим число 106, сумму кодов символов '4' и '6'

проделаем аналогичную работу с прошлым примером, заменив

```
mov eax, '6'
```

```
mov ebx, '4'
```

на

```
mov eax, 6
```

```
mov ebx, 4
```

и, запустив файл, получим число 10, сумму 4 и 6.

(рис. 5.4)

Также заменим в исходном файле функцию `iprintLF` на `iprint`, и увидим, что из-за этого строка не переводится после результата.

(рис. 5.5)

Шаг 4

В новый файл `lab7-3.asm` введём текст из листинга 7.3, для нахождения выражения $((5 \times 2) + 3) / 3$

Создаём файл и получаем ответ:

(рис. 5.6)

Изменим код файла для вычисления выражения $((4 \times 6) + 2) / 5$

(рис. 5.7)

Результат:

(рис. 5.8)

Шаг 5

вводим в `variant.asm` текст программы, проверяем его работу:

(рис. 5.9)

Убедимся в правильности результата, разделим 1132226495 на 20 на калькуляторе, получим 56611324 целых, 15 в остатке, добавим к остатку 1, это равняется полученному номеру варианта

4 Ответы на вопросы 1-7

1. переменная `rem` отвечает за сообщение 'Ваш вариант:'

```
rem: DB 'Ваш вариант: ',0
```

```
,
```

```
mov eax, msg
```

```
call sprintf
```

Отвечает за вывод сообщения 'ваш вариант: [вариант]'

2.

Данные функции используются для ввода значения переменной `X` с клавиатуры и сохранения введенных данных

3.

Данная инструкция используется для преобразования кода символа ASCII в число

4.

```
mov ebx,20
```

```
div ebx
```

```
inc edx
```


5.

В регистр edx

6.

Для увеличения значения edx на 1.

7.

Результат вычислений = номер варианта

```
mov eax,edx
```

```
call iprintLF
```

5 Задание для самостоятельной работы

Шаг 1

Создадим файл var16.asm, в которой решим выражение $(10 \cdot x - 5)^2$, в зависимости от различных X. За основу возьмём Variant.asm

Рассмотрим принцип работы нашего кода (Также см. Комментарии в самом коде)

```
mov eax, msg
call sprintf
mov ecx, x
mov edx, 80
call sread
mov eax, x
call atoi
```

отвечает за запись переменной X

Далее, умножим X (eax) на 10 (ebx)

вычтем из X (eax) 5

и умножим на X (eax)

Запишем результат в edi, выведем на экран с помощью iprintLF

(рис. 5.10)

```
[dnthetinin@fedora ~]$ mkdir ~/work/arch-pc/lab07
[dnthetinin@fedora ~]$ cd ~/work/arch-pc/lab07
[dnthetinin@fedora lab07]$ touch lab7-1.asm
[dnthetinin@fedora lab07]$ nasm -f elf lab7-1.asm
[dnthetinin@fedora lab07]$ ld -m elf_i386 -o lab7-1 lab7-1.o
[dnthetinin@fedora lab07]$ ./lab7-1
j
```

Рис. 5.1: lab7-1

```
[dnthetinin@fedora lab07]$ nasm -f elf lab7-1.asm
[dnthetinin@fedora lab07]$ ld -m elf_i386 -o lab7-1 lab7-1.o
[dnthetinin@fedora lab07]$ ./lab7-1
```

Рис. 5.2: lab7-1

```
[dnthetinin@fedora lab07]$ touch lab7-2.asm
[dnthetinin@fedora lab07]$ nasm -f elf lab7-2.asm
[dnthetinin@fedora lab07]$ ld -m elf_i386 -o lab7-2 lab7-2.o
[dnthetinin@fedora lab07]$ ./lab7-2
106
```

Рис. 5.3: lab7-2

```
[dnthetinin@fedora lab07]$ nasm -f elf lab7-2.asm
[dnthetinin@fedora lab07]$ ld -m elf_i386 -o lab7-2 lab7-2.o
[dnthetinin@fedora lab07]$ ./lab7-2
10
```

Рис. 5.4: lab7-2

```
[dnthetinin@fedora lab07]$ nasm -f elf lab7-2.asm
[dnthetinin@fedora lab07]$ ld -m elf_i386 -o lab7-2 lab7-2.o
[dnthetinin@fedora lab07]$ ./lab7-2
10[dnthetinin@fedora lab07]$ touch lab7-3.asm
```

Рис. 5.5: lab7-2

```
[dnthetinin@fedora lab07]$ nasm -f elf lab7-3.asm
[dnthetinin@fedora lab07]$ ld -m elf_i386 -o lab7-3 lab7-3.o
[dnthetinin@fedora lab07]$ ./lab7-3
Результат: 4
Остаток от деления: 1
```

Рис. 5.6: работа файла lab7-3

Открыть ▼

+

lab7-3.asm
~/work/arch-pc/lab07

```

; (5*2 + 3)/3
; (4*6 + 2)/5
%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=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 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. 5.7: Код новой lab7-3.asm



```
[dnthetinin@fedora lab07]$ nasm -f elf lab7-3.asm
[dnthetinin@fedora lab07]$ ld -m elf_i386 -o lab7-3 lab7-3.o
[dnthetinin@fedora lab07]$ ./lab7-3
Результат: 5
Остаток от деления: 1
[dnthetinin@fedora lab07]$
```

Рис. 5.8: Результат новой lab7-3

```
[dnthetinin@fedora lab07]$ ./variant
Введите No студенческого билета:
1132226495
Ваш вариант: 16
[dnthetinin@fedora lab07]$
```

Рис. 5.9: Полученный вариант

Открыть  • var16.asm
~\work\arch-pc\lab07

Стр. 21, Поз. 1  

report.md • var16.asm

```
1 ; (10*x-5)^2
2 ; x1=3
3 ; x2=1
4
5 %include 'in_out.asm' ; подключение внешнего файла
6
7 SECTION .data
8 msg: DB 'Введите X: ',0
9 div: DB 'Результат: ',0
10 SECTION .bss
11 x: RESB 80
12 SECTION .text
13 GLOBAL _start
14 _start:
15
16 ;Вычисление выражения
17
18 mov eax, msg
19 call sprintf
20 mov ecx, x
21 mov edx, 80
22 call sread ; запись введённого значения
23 mov eax,x ; x в переменную eax
24 call atoi ; вызов подпрограммы преобразования ASCII кода в число
25
26 mov ebx,10 ; ebx=10
27
28 mul ebx ; EAX * 10
29 sub eax,5 ; EAX - 5
30 mul eax ; EAX * EAX
31
32 mov edi,eax ; запись результата вычисления в 'edi'
33
34 ; ---- Вывод результата на экран
35
36 mov eax,div ; вызов подпрограммы печати
37 call sprint ; сообщения 'Результат: '
38 mov eax,edi ; вызов подпрограммы печати значения
39 call iprintf ; из 'edi' в виде символов
40
41 call quit ; вызов подпрограммы завершения
```

Рис. 5.10: код var16

6 Выводы

Я смог написать базовую программу вычисления функции на языке ассемблера
NASM