

Отчёт по лабораторной работе №6

Арифметические операции в NASM.

Степан Михайлович Токаев

Содержание

1	Цель работы	1
2	Выполнение лабораторной работы.....	1
3	Ответы на вопросы по программе.....	9
4	Выполнение заданий для самостоятельной работы.....	9
5	Вывод.....	11

1 Цель работы

Освоение арифметических инструкций языка ассемблера NASM.

2 Выполнение лабораторной работы

1. Создадим каталог для программ лабораторной работы No 6, перейдём в него и создадим файл lab6-1.asm

```
(smtokaev@ smtokaev)~  
$ mkdir ~/work/study/2022-2023/Архитектура\ компьютера/arch-pc/lab06  
  
(smtokaev@ smtokaev)~  
$ cd ~/work/study/2022-2023/Архитектура\ компьютера/arch-pc/lab06  
  
(smtokaev@ smtokaev)~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab06  
$ touch lab6-1.asm
```

Рис. 1: Создание файла

2. Введём в файл lab6-1.asm текст программы из листинга 7.1

```

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

```

Рис. 2: Ввод теста

3. Создадим исполняемый файл и запустим его.

```

(smtokaev@сmtokaev)-[~/.../2022-2023/Архитектура компьютера/arch-рс/lab06]
$ nasm -f elf lab6-1.asm

(smtokaev@сmtokaev)-[~/.../2022-2023/Архитектура компьютера/arch-рс/lab06]
$ ld -m elf_i386 -o lab6-1 lab6-1.o

(smtokaev@сmtokaev)-[~/.../2022-2023/Архитектура компьютера/arch-рс/lab06]
$ ./lab6-1
j

```

Рис. 3: Создание и запуск файла

4. Изменим текст программы и вместо символов, запишем в регистры числа.

```

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

```

Рис. 4: Изменение текста

5. Создадим исполняемый файл и запустим его.

```

(smtokaev@сmtokaev)-[~/.../2022-2023/Архитектура компьютера/arch-рс/lab06]
$ nasm -f elf lab6-1.asm

(smtokaev@сmtokaev)-[~/.../2022-2023/Архитектура компьютера/arch-рс/lab06]
$ ld -m elf_i386 -o lab6-1 lab6-1.o

(smtokaev@сmtokaev)-[~/.../2022-2023/Архитектура компьютера/arch-рс/lab06]
$ ./lab6-1

```

Рис. 5: Создание и запуск файла

Данный символ не выводится на экран

6. Создадим файл lab6-2.asm и введём в него текст программы из листинга 6.2.

```

(smtokaev@сmtokaev)-[~/.../2022-2023/Архитектура компьютера/arch-рс/lab06]
$ touch lab6-2.asm

```

Рис. 6: Создание файла

```

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|

```

Рис. 7: Текст программы в файле

7. Создайте исполняемый файл и запустите его.

```

(smtokaev@сmtokaev)-[~/.../2022-2023/Архитектура компьютера/arch-pc/lab06]
$ nasm -f elf lab6-2.asm

(smtokaev@сmtokaev)-[~/.../2022-2023/Архитектура компьютера/arch-pc/lab06]
$ ld -m elf_i386 -o lab6-2 lab6-2.o

(smtokaev@сmtokaev)-[~/.../2022-2023/Архитектура компьютера/arch-pc/lab06]
$ ./lab6-2
106

```

Рис. 8: Создание и запуск файла

8. Заменяем символы на числа.

```

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|

```

Рис. 9: Замена строк

9. Создадим исполняемый файл и запустим его. В результате получили число 10.

```

(smtokaev@сmtokaev)-[~/.../2022-2023/Архитектура компьютера/arch-pc/lab06]
$ nasm -f elf lab6-2.asm

(smtokaev@сmtokaev)-[~/.../2022-2023/Архитектура компьютера/arch-pc/lab06]
$ ld -m elf_i386 -o lab6-2 lab6-2.o

(smtokaev@сmtokaev)-[~/.../2022-2023/Архитектура компьютера/arch-pc/lab06]
$ ./lab6-2
10

```

Рис. 10: Создание и запуск файла

10. Заменяем функцию `iprintLF` на `iprint`, создадим исполняемый файл и запустим его. Вывод функций `iprintLF` и `iprint` отличается наличием перевода 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

```

Рис. 11: Изменение текста

```

(smtokaev@сmtokaev)-[~/.../2022-2023/Архитектура компьютера/arch-pc/lab06]
$ nasm -f elf lab6-2.asm

(smtokaev@сmtokaev)-[~/.../2022-2023/Архитектура компьютера/arch-pc/lab06]
$ ld -m elf_i386 -o lab6-2 lab6-2.o

(smtokaev@сmtokaev)-[~/.../2022-2023/Архитектура компьютера/arch-pc/lab06]
$ ./lab6-2
10

```

Рис. 12: Создание и запуск файла

11. Создадим файл `lab6-3.asm`.

```

(smtokaev@сmtokaev)-[~/.../2022-2023/Архитектура компьютера/arch-pc/lab06]
$ touch lab6-3.asm

```

Рис. 13: Создание файла

12. Введём в `lab6-3.asm` текст программы из листинга 7.3

```

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

```

Рис. 14: Ввод текста

13. Создадим исполняемый файл и запустим его.

```

(smtokaev@сmtokaev)-[~/.../2022-2023/Архитектура компьютера/arch-pc/lab06]
$ nasm -f elf lab6-3.asm

(smtokaev@сmtokaev)-[~/.../2022-2023/Архитектура компьютера/arch-pc/lab06]
$ ld -m elf_i386 -o lab6-3 lab6-3.o

(smtokaev@сmtokaev)-[~/.../2022-2023/Архитектура компьютера/arch-pc/lab06]
$ ./lab6-3
Результат: 4
Остаток от деления: 1

```

Рис. 15: Ссоздание файла

14. Изменим текст программы для вычисления выражения $f(x) = (4 * 6 + 2)/5$.
Создадим исполняемый файл и проверим его работу.


```

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

```

Рис. 16: Изменение текста

```

(smtokaev@сmtokaev)-[~/2022-2023/Архитектура компьютера/arch-pc/lab06]
$ nasm -f elf lab6-3.asm

(smtokaev@сmtokaev)-[~/2022-2023/Архитектура компьютера/arch-pc/lab06]
$ ld -m elf_i386 -o lab6-3 lab6-3.o

(smtokaev@сmtokaev)-[~/2022-2023/Архитектура компьютера/arch-pc/lab06]
$ ./lab6-3
Результат: 5
Остаток от деления: 1

```

Рис. 17: Создание и проверка файла

15. Создадим файл variant.asm для вычисления варианта задания по номеру студенческого билета.

```
(smtokaev@сmtokaev)-[~/.../2022-2023/Архитектура компьютера/arch-рс/lab06]  
$ touch variant.asm
```

Рис. 18: Создание файла

16. введём в файл variant.asm текст программы из листинга 7.4

```
1 %include 'in_out.asm'  
2 SECTION .data  
3 msg: DB 'Введите No студенческого билета: ',0  
4 rem: DB 'Ваш вариант: ',0  
5 SECTION .bss  
6 x: RESB 80  
7 SECTION .text  
8 GLOBAL _start  
9 _start:  
10 mov eax, msg  
11 call sprintf  
12 mov ecx, x  
13 mov edx, 80  
14 call sread  
15 mov eax, x ; вызов подпрограммы преобразования  
16 call atoi ; ASCII кода в число, `eax=x`  
17 xor edx, edx  
18 mov ebx, 20  
19 div ebx  
20 inc edx  
21 mov eax, rem  
22 call sprintf  
23 mov eax, edx  
24 call iprintLF  
25 call quit
```

Рис. 19: Ввод текста

17. Создадим исполняемый файл и запустим его. Результат работы программы вычислил номер варианта - 8.


```
(smtokaev@smtokeev)-[~/.../2022-2023/Архитектура компьютера/arch-pc/lab06]
$ nasm -f elf variant.asm

(smtokaev@smtokeev)-[~/.../2022-2023/Архитектура компьютера/arch-pc/lab06]
$ ld -m elf_i386 -o variant variant.o

(smtokaev@smtokeev)-[~/.../2022-2023/Архитектура компьютера/arch-pc/lab06]
$ ./variant
Введите No студенческого билета:
1132220819
Ваш вариант: 20
```

Рис. 20: Создание и запуск файла

3 Ответы на вопросы по программе

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

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

2. Инструкция `mov esx, x` используется, чтобы положить адрес вводимой строки в регистр `esx` `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
```

4 Выполнение заданий для самостоятельной работы

1. Создадим файл `variant-2.asm`

```
(smtokaev@smtokeev)-[~/.../2022-2023/Архитектура компьютера/arch-pc/lab06]
$ touch variant-2.asm
```

Рис. 21: Создание файла

2. Вводим текст программы для вычисления значения выражения из варианта 20

```

1 %include 'in_out.asm' ; подключение внешнего файла
2 SECTION .data
3 div: DB 'Результат: ',0
4 rem1: DB 'Введите x: ',0
5 rem2: DB 'x^3 * 1/3 + 21',0
6 SECTION .bss
7 x: RESB 80
8 SECTION .text
9 GLOBAL _start
10 _start:
11
12 mov eax, rem2
13 call sprintf
14 mov eax, rem1
15 call sprintf
16 mov ecx, x
17 mov edx, 80
18 call sread
19 mov eax,x ; вызов подпрограммы преобразования
20 call atoi
21
22 mov ebx,eax ;
23 mul ebx ; EAX=EAX*EBX
24 mul ebx
25 mov ebx, 1
26 mul ebx
27 xor edx,edx ; обнуляем EDX для корректной работы div
28 mov ebx,3 ;
29 div ebx ;
30 mov edi,eax ; запись результата вычисления в 'edi'
31
32 add edi, 21
33 ; — Вывод результата на экран
34 mov eax,div ; вызов подпрограммы печати
35 call sprintf ; сообщения 'Результат: '
36 mov eax,edi ; вызов подпрограммы печати значения
37 call iprintLF ; из 'edi' в виде символов
38 call quit ; вызов подпрограммы завершения

```

Рис. 22: Ввод текста

3. Создадим исполняемый файл и проверим его работу для значений $x_1 = 1$ и $x_2 = 3$.

```
(smtokaev@сmtokaev)-[~/.../2022-2023/Архитектура компьютера/arch-pc/lab06]
$ nasm -f elf variant-2.asm

(smtokaev@сmtokaev)-[~/.../2022-2023/Архитектура компьютера/arch-pc/lab06]
$ ld -m elf_i386 -o variant-2 variant-2.o

(smtokaev@сmtokaev)-[~/.../2022-2023/Архитектура компьютера/arch-pc/lab06]
$ ./variant-2
x^3 * 1/3 + 21
Введите x: 1
Результат: 21

(smtokaev@сmtokaev)-[~/.../2022-2023/Архитектура компьютера/arch-pc/lab06]
$ ./variant-2
x^3 * 1/3 + 21
Введите x: 3
Результат: 30
```

Рис. 23: Создание и проверка файла

5 Вывод

При выполнении данной лабораторной работы я освоил арифметические инструкции языка ассемблера NASM. Здесь кратко описываются итоги проделанной работы.