

ОТЧЕТ по лабораторной работе №6

Полякова Юлия Александровна

Содержание

1	Цель работы	5
2	Результаты выполнения лабораторной работы	6
3	Результаты выполнения заданий для самостоятельной работы	15
4	Вывод	18

Список иллюстраций

2.1	Создание каталога и lab6-1.asm	6
2.2	lab6-1.asm из листинга 6.1.	6
2.3	Запуск lab6-1.asm из листинга 6.1.	7
2.4	Измененный lab6-1.asm	7
2.5	Запуск измененного lab6-1.asm	8
2.6	Символ переноса строки	8
2.7	lab6-2.asm из листинга 6.2.	9
2.8	Запуск lab6-2.asm из листинга 6.2.	9
2.9	Измененный lab6-2.asm	10
2.10	Запуск измененного lab6-2.asm	10
2.11	Работа файла с iprint	10
2.12	lab6-3.asm из листинга 6.3.	11
2.13	Запуск lab6-3.asm из листинга 6.3.	11
2.14	Измененный lab6-3.asm	12
2.15	Запуск измененного lab6-3.asm	12
2.16	variant.asm из листинга 6.4.	13
2.17	Запуск variant.asm из листинга 6.4.	13
3.1	Программа для вычисления $(2 + x)^2$	16
3.2	Проверка работы при $x_1 = 2$ и $x_2 = 8$	17

Список таблиц

1 Цель работы

Освоение арифметических инструкций языка ассемблера NASM.

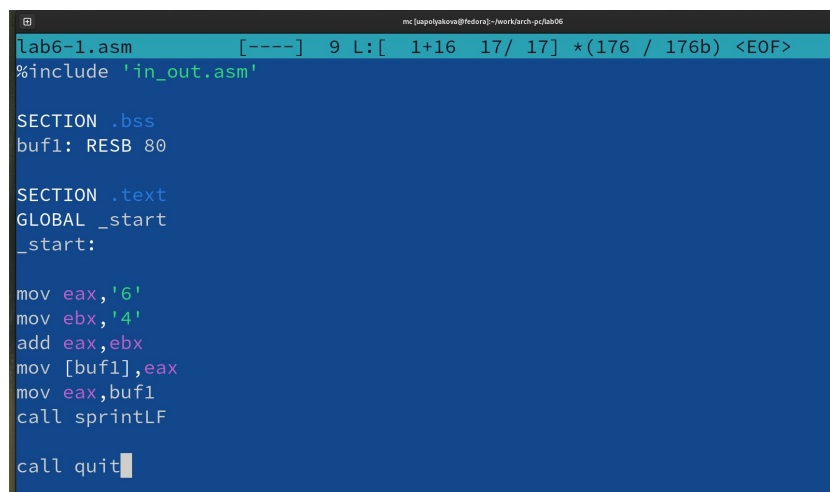
2 Результаты выполнения лабораторной работы

1. Создаем каталог для программ лабораторной №6, переходим в него и создаем файл lab6-1.asm (Рис. 1).

```
uapolyakova@fedora:~$ mkdir ~/work/arch-pc/lab06
uapolyakova@fedora:~$ cd ~/work/arch-pc/lab06
uapolyakova@fedora:~/work/arch-pc/lab06$ touch lab6-1.asm
uapolyakova@fedora:~/work/arch-pc/lab06$ ls
lab6-1.asm
uapolyakova@fedora:~/work/arch-pc/lab06$
```

Рис. 2.1: Создание каталога и lab6-1.asm

2. Записываем в файл программу из листинга 6.1. (Рис. 2).



```
lab6-1.asm [----] 9 L: [ 1+16 17/ 17] *(176 / 176b) <EOF>
#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. 2.2: lab6-1.asm из листинга 6.1.

3. Создаем исполняемый файл и запускаем его (Рис. 3).

```
uapolyakova@fedora:~/work/arch-pc/lab06$ nasm -f elf lab6-1.asm
uapolyakova@fedora:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-1 lab6-1.o
ld: не распознан режим эмуляции: elf_i386
Поддерживаемые эмуляции: elf_x86_64 elf32_x86_64 elf_i386 elf_iamcu i386pep i386pe elf64bpf
uapolyakova@fedora:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-1 lab6-1.o
uapolyakova@fedora:~/work/arch-pc/lab06$ ./lab6-1
j
uapolyakova@fedora:~/work/arch-pc/lab06$
```

Рис. 2.3: Запуск lab6-1.asm из листинга 6.1.

4. Изменяем текст программы, вместо символов записываем в регистры числа (Рис. 4)

```
lab6-1.asm [----] 9 L: [ 1+10 11/
#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. 2.4: Измененный lab6-1.asm

5. Создаем исполняемый файл и запускаем его (Рис. 5).

```

uapolyakova@fedora:~/work/arch-pc/lab06$ nasm -f elf lab6-1.asm
uapolyakova@fedora:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-1 lab6-1.o
uapolyakova@fedora:~/work/arch-pc/lab06$ ./lab6-1

uapolyakova@fedora:~/work/arch-pc/lab06$

```

Рис. 2.5: Запуск измененного lab6-1.asm

По таблице ASCII это символ переноса строки \n (Рис. 6).

ASCII Table

ASCII Table - Complete ASCII code chart with characters. Also, it contains decimal, hexadecimal, binary, and HTML values.

Dec	Hex	Binary	HTML	Char	Description
0	00	00000000	�	NUL	Null
1	01	00000001		SOH	Start of Header
2	02	00000010		STX	Start of Text
3	03	00000011		ETX	End of Text
4	04	00000100		EOT	End of Transmission
5	05	00000101		ENQ	Enquiry
6	06	00000110		ACK	Acknowledge
7	07	00000111		BEL	Bell
8	08	00001000		BS	Backspace
9	09	00001001			HT	Horizontal Tab
10	0A	00001010	
	LF	Line Feed
11	0B	00001011		VT	Vertical Tab
12	0C	00001100		FF	Form Feed

Рис. 2.6: Символ переноса строки

Он отображается при выводе на экран, так как видно, что после запуска программы образуются две пустые строки.

6. Создаем lab6-2.asm по листингу 6.2. (Рис. 7).


```

lab6-2.asm                                [-M--]
#include 'in_out.asm'

SECTION .text
GLOBAL _start
_start:

mov  eax, '6'
mov  ebx, '4'
add  eax, ebx
call iprintLF

call quit

```

Рис. 2.7: lab6-2.asm из листинга 6.2.

7. Создаем исполняемый файл и запускаем его (Рис. 8).

```

uapolyakova@fedora:~/work/arch-pc/lab06$ touch lab6-2.asm
uapolyakova@fedora:~/work/arch-pc/lab06$ mc

uapolyakova@fedora:~/work/arch-pc/lab06$ nasm -f elf lab6-2.asm
uapolyakova@fedora:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-2 lab6-2.o
uapolyakova@fedora:~/work/arch-pc/lab06$ ./lab6-2
106
uapolyakova@fedora:~/work/arch-pc/lab06$

```

Рис. 2.8: Запуск lab6-2.asm из листинга 6.2.

8. Изменяем символы на числа.

```

lab6-2.asm [----] 9
#include 'in_out.asm'

SECTION .text
GLOBAL _start
_start:

mov eax,6
mov ebx,4
add eax,ebx
call iprintLF

call quit

```

Рис. 2.9: Измененный lab6-2.asm

9. Создаем исполняемый файл и запускаем его (Рис. 10).

```

uapolyakova@fedora:~/work/arch-pc/lab06$ nasm -f elf lab6-2.asm
uapolyakova@fedora:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-2 lab6-2.o
uapolyakova@fedora:~/work/arch-pc/lab06$ ./lab6-2
10
uapolyakova@fedora:~/work/arch-pc/lab06$

```

Рис. 2.10: Запуск измененного lab6-2.asm

В результате выведется число 10, а не символ, кодом которого является это число.

10. Заменяем iprintLF на iprint, создаем исполняемый файл и запускаем его (Рис. 11).

```

uapolyakova@fedora:~/work/arch-pc/lab06$ nasm -f elf lab6-2.asm
uapolyakova@fedora:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-2 lab6-2.o
uapolyakova@fedora:~/work/arch-pc/lab06$ ./lab6-2
10uapolyakova@fedora:~/work/arch-pc/lab06$

```

Рис. 2.11: Работа файла с iprint

Функция `iprint` выводит без переноса строки.

11. Создаем `lab6-3.asm` по листингу 6.3. (Рис. 12).

```
lab6-3.asm [----] 0 L: [ 1+16 17/ 31] *(473 / 715)
%include 'in_out.asm'
SECTION .data
div: DB 'Результат: ',0
rem: DB 'Остаток от деления: ',0

SECTION .text
GLOBAL _start
_start:
; Вычисление выражения (5*2+3)/3
mov eax,5 ; eax = 5
mov ebx,2 ; ebx = 2
mul ebx ; eax *= ebx
add eax,3 ; eax += 3
xor edx,edx ; обнуляем edx для корректной работы div
mov ebx,3 ; ebx = 3
div ebx ; eax /= 3, edx = остаток от деления

mov edi,eax ; запись результата вычисления в 'edi'

; Вывод результата на экран
mov eax,div
call sprint
mov eax,edi
call iprintLF
```

Рис. 2.12: `lab6-3.asm` из листинга 6.3.

12. Создаем исполняемый файл и запускаем его (Рис. 13).

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

Рис. 2.13: Запуск `lab6-3.asm` из листинга 6.3.

13. Изменяем текст для вычисления $(4*6+2)/5$ (Рис. 14).

```

; Вычисление выражения (4*6+2)/5
mov eax,4      ; eax = 4
mov ebx,6      ; ebx = 6
mul ebx        ; eax *= ebx
add eax,2      ; eax += 2
xor edx,edx    ; обнуляем edx для корректной работы div
mov ebx,5      ; ebx = 5
div ebx        ; eax /= 5, edx = остаток от деления

mov edi,eax    ; запись результата вычисления в 'edi'

```

Рис. 2.14: Измененный lab6-3.asm

14. Создаем исполняемый файл и запускаем его (Рис. 15).

```

uapolyakova@fedora:~/work/arch-pc/lab06$ nasm -f elf lab6-3.asm
uapolyakova@fedora:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-3 lab6-3.o
uapolyakova@fedora:~/work/arch-pc/lab06$ ./lab6-3
Результат: 5
Остаток от деления: 1
uapolyakova@fedora:~/work/arch-pc/lab06$

```

Рис. 2.15: Запуск измененного lab6-3.asm

15. Создаем variant.asm по листингу 6.4. (Рис. 16).

```

variant.asm      [----] 13 L:[ 1+ 8 9/ 33] *(184
%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
call atoi

xor edx,edx
mov ebx,20
div ebx

```

Рис. 2.16: variant.asm из листинга 6.4.

16. Создаем исполняемый файл и запускаем его (Рис. 17).

```

uapolyakova@fedora:~/work/arch-pc/lab06$ touch variant.asm
uapolyakova@fedora:~/work/arch-pc/lab06$ mc

uapolyakova@fedora:~/work/arch-pc/lab06$ nasm -f elf variant.asm
uapolyakova@fedora:~/work/arch-pc/lab06$ ld -m elf_i386 -o variant variant.o
uapolyakova@fedora:~/work/arch-pc/lab06$ ./variant
Введите № студенческого билета:
1132243102
Ваш вариант: 3
uapolyakova@fedora:~/work/arch-pc/lab06$

```

Рис. 2.17: Запуск variant.asm из листинга 6.4.

Проверим. Остаток от деления последних цифр 02 на 20 это 2. 2 + 1 это 3. Следовательно вариант 3.

Ответы на вопросы:

1. Строки

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

2. Для записи адреса переменной в 'EAX', записи длины вводимого сообщения в 'EBX' и вызова подпрограммы ввода сообщения.

3. Для преобразования символов ASCII кода в числа.

4. Строки

```
xor edx,edx  
mov ebx,20  
div ebx  
inc edx
```

5. В регистр edx.

6. Чтобы прибавить единицу (сделать инкремент) к остатку от деления на 20.

7. Строки

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

3 Результаты выполнения заданий для самостоятельной работы

1. Вариант 3. Программа для вычисления $(2 + x)^2$ в файле samrab.asm (рис. 18).

```

1 %include 'in_out.asm'
2 SECTION .data
3 msg: DB 'Введите x: ',0
4 rem: DB 'Результат выражения (2 + x)^2: ',0
5
6 SECTION .bss
7 x: RESB 80
8
9 SECTION .text
10 GLOBAL _start
11 _start:
12
13 mov eax,msg
14 call sprintf
15
16 mov ecx,x
17 mov edx,80
18 call sread
19
20 mov eax,x
21 call atoi
22
23 add eax,2
24 mul eax
25
26 mov edi,eax
27
28 mov eax,rem
29 call sprintf
30 mov eax,edi
31 call iprintLF
32
33 call quit

```

Рис. 3.1: Программа для вычисления $(2 + x)^2$

2. Создаем исполняемый файл и проверяем его работу (рис. 19).


```
uapolyakova@fedora:~/work/arch-pc/lab06$ nasm -f elf samrab.asm
uapolyakova@fedora:~/work/arch-pc/lab06$ ld -m elf_i386 -o samrab samrab.o
uapolyakova@fedora:~/work/arch-pc/lab06$ ./samrab
Введите x:
2
Результат выражения (2 + x)^2: 16
uapolyakova@fedora:~/work/arch-pc/lab06$ ./samrab
Введите x:
8
Результат выражения (2 + x)^2: 100
uapolyakova@fedora:~/work/arch-pc/lab06$
```

Рис. 3.2: Проверка работы при $x_1 = 2$ и $x_2 = 8$

4 Вывод

Были освоены арифметические инструкции языка ассемблера NASM.