

Лабораторная работа №6

Воинов Кирилл Викторович

Содержание

1	Цель работы	4
2	Выполнение лабораторной работы	5
3	Ответы на вопросы	13
4	Задание для самостоятельной работы	15
5	Вывод	18

Список иллюстраций

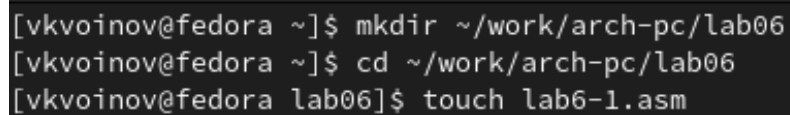
2.1	Создание каталога и файла	5
2.2	Текст программы	6
2.3	Вывод из файла lab6-1.asm	6
2.4	Изменённый текст программы	7
2.5	Вывод из отредактированного файла lab6-1.asm	7
2.6	Создание файла lab6-2.asm	7
2.7	Текст программы	8
2.8	Запуск исполняемого файла	8
2.9	Изменённый текст программы	9
2.10	Вывод из отредактированного файла lab6-2.asm	9
2.11	Вывод при замене iprintLF на iprint	9
2.12	Создание файла	10
2.13	Вывод программы	10
2.14	Изменённый текст	11
2.15	Вывод изменённой программы	11
2.16	Создание файла variant.asm	12
2.17	Вывод программы	12
4.1	Текст программы	16
4.2	Вывод программы	17

1 Цель работы

Освоение арифметических инструкций языка ассемблера NASM.

2 Выполнение лабораторной работы

1. Создаю каталог для программ лабораторной работы №6, перехожу в него и создаю файл lab6-1.asm. (рис. 2.1)



```
[vkvoinov@fedora ~]$ mkdir ~/work/arch-pc/lab06  
[vkvoinov@fedora ~]$ cd ~/work/arch-pc/lab06  
[vkvoinov@fedora lab06]$ touch lab6-1.asm
```

Рис. 2.1: Создание каталога и файла

2. Ввожу в файл lab6-1.asm текст программы из листинга 6.1, создаю исполняемый файл и запускаю его.(рис. 2.2) и (рис. 2.3)

```

%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: Текст программы

```

[vkvoinov@fedora lab06]$ nasm -f elf lab6-1.asm
[vkvoinov@fedora lab06]$ ld -m elf_i386 -o lab6-1 lab6-1.o
[vkvoinov@fedora lab06]$ ./lab6-1
j
[vkvoinov@fedora lab06]$

```

Рис. 2.3: Вывод из файла lab6-1.asm

3. Изменяю текст программы и вместо символов, записываю в регистры числа.(рис. 2.4)

```
SECTION .text
GLOBAL _start
_start:
mov eax,6
mov ebx,4
add eax,ebx
mov [buf1],eax
mov eax,buf1
call sprintf
call quit
```

Рис. 2.4: Изменённый текст программы

Создаю исполняемый файл и запускаю его.(рис. 2.5)

```
[vkvoinov@fedora lab06]$ nasm -f elf lab6-1.asm
[vkvoinov@fedora lab06]$ ld -m elf_i386 -o lab6-1 lab6-1.o
[vkvoinov@fedora lab06]$ ./lab6-1
```

Рис. 2.5: Вывод из отредактированного файла lab6-1.asm

Этот код соответствует символу STX. Он не отображается на экран.

4. Создаю файл lab6-2.asm в каталоге ~/work/arch-pc/lab06 и ввожу в него текст программы из листинга 6.2. (рис.@fig:006) и (рис. 2.7)

```
[vkvoinov@fedora lab06]$ touch ~/work/arch-pc/lab06/lab6-2.asm
[vkvoinov@fedora lab06]$
```

Рис. 2.6: Создание файла lab6-2.asm

```

%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: Текст программы

Создаю исполняемый файл и запускаю его. (рис. 2.8)

```

[vkvoinov@fedora lab06]$ nasm -f elf lab6-2.asm
[vkvoinov@fedora lab06]$ ld -m elf_i386 -o lab6-2 lab6-2.o
[vkvoinov@fedora lab06]$ ./lab6-2
106
[vkvoinov@fedora lab06]$

```

Рис. 2.8: Запуск исполняемого файла

5. Аналогично предыдущему примеру изменяю символы на числа.(рис. 2.9)


```

%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: Изменённый текст программы

Создаю исполняемый файл и запускаю его.(рис. 2.10)

```

[vkvoinov@fedora lab06]$ nasm -f elf lab6-2.asm
[vkvoinov@fedora lab06]$ ld -m elf_i386 -o lab6-2 lab6-2.o
[vkvoinov@fedora lab06]$ ./lab6-2
10
[vkvoinov@fedora lab06]$

```

Рис. 2.10: Вывод из отредактированного файла lab6-2.asm

Заменяю функцию iprintLF на iprint. Создаю исполняемый файл и запускаю его.(рис. 2.11)

```

[vkvoinov@fedora lab06]$ nasm -f elf lab6-2.asm
[vkvoinov@fedora lab06]$ ld -m elf_i386 -o lab6-2 lab6-2.o
[vkvoinov@fedora lab06]$ ./lab6-2
10[vkvoinov@fedora lab06]$

```

Рис. 2.11: Вывод при замене iprintLF на iprint

Вывод функций iprintLF и iprint отличается тем, что в первом случае после вывода происходит переход на новую строку, а во втором нет.

6. Создаю файл lab6-3.asm в каталоге ~/work/arch-pc/lab06.(рис. 2.12)

```
[vkvoinov@fedora lab06]$ touch ~/work/arch-pc/lab06/lab6-3.asm  
[vkvoinov@fedora lab06]$
```

Рис. 2.12: Создание файла

Ввожу в него текст программы из листинга 6.3, создаю исполняемый файл и запускаю его.(рис. 2.13)

```
[vkvoinov@fedora lab06]$ nasm -f elf lab6-3.asm  
[vkvoinov@fedora lab06]$ ld -m elf_i386 -o lab6-3 lab6-3.o  
[vkvoinov@fedora lab06]$ ./lab6-3  
Результат: 4  
Остаток от деления: 1  
[vkvoinov@fedora lab06]$
```

Рис. 2.13: Вывод программы

Изменяю текст программы для вычисления выражения $f(x)=(4*6+2)/5$. (рис. 2.14)

```

%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. 2.14: Изменённый текст

Создаю исполняемый файл и проверяю его работу. (рис. 2.15)

```

[vkvoinov@fedora lab06]$ nasm -f elf lab6-3.asm
[vkvoinov@fedora lab06]$ ld -m elf_i386 -o lab6-3 lab6-3.o
[vkvoinov@fedora lab06]$ ./lab6-3
Результат: 5
Остаток от деления: 1
[vkvoinov@fedora lab06]$ 

```

Рис. 2.15: Вывод изменённой программы

7. Создаю файл variant.asm в каталоге ~/work/arch-pc/lab06. (рис. 2.16)

```
[vkvoinov@fedora lab06]$ touch ~/work/arch-pc/lab06/variant.asm
```

Рис. 2.16: Создание файла variant.asm

Ввожу в него текст программы из листинга 6.3, создаю исполняемый файл и запускаю его.(рис. 2.17)

```
[vkvoinov@fedora lab06]$ nasm -f elf variant.asm
[vkvoinov@fedora lab06]$ ld -m elf_i386 -o variant variant.o
[vkvoinov@fedora lab06]$ ./variant
Введите No студенческого билета:
1132236017
Ваш вариант: 18
[vkvoinov@fedora lab06]$
```

Рис. 2.17: Вывод программы

Остаток деления 1132236017 на 20 это 17 и $17+1=18$.

3 Ответы на вопросы

1. Какие строки листинга 6.4 отвечают за вывод на экран сообщения 'Ваш вариант:'?

Строки:

```
rem: DB 'Ваш вариант:',0  
mov eax,rem  
call sprint
```

2. Для чего используются следующие инструкции?

```
mov ecx, x  
mov edx, 80  
call sread
```

Для ввода значения с длиной 80, передачи этого значения в x.

3. Для чего используется инструкция "call atoi"?

Для преобразования ASCII кода в число.

4. Какие строки листинга 6.4 отвечают за вычисления варианта?

Строки:

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

5. В какой регистр записывается остаток от деления при выполнении инструкции “div ebx”?

В регистр edx.

6. Для чего используется инструкция “inc edx”?

Для прибавления единицы.

7. Какие строки листинга 6.4 отвечают за вывод на экран результата вычисления?

Строки:

```
mov eax,edx
```

```
call iprintLF
```

4 Задание для самостоятельной работы

1. Пишу программу вычисления выражения $y=f(x)$. Вариант 18: $f(x)=3(x+10)-20$. (рис. 4.1)

```

%include 'in_out.asm'
SECTION .data
msv: DB 'f(x)=3(x+10)-20 ',0
msg: DB 'Введите x: ',0
div: DB 'Результат: ',0
SECTION .bss
x: RESB 80
SECTION .text
GLOBAL _start
_start:
; ---- Ввод x
mov eax, msv
call sprintLF
mov eax, msg
call sprintLF
mov ecx, x
mov edx, 80
call sread
mov eax, x
call atoi
; ---- Вычисление выражения
add eax, 10
mov ebx, 3
mul ebx
add eax, -20
mov edi, eax
; ---- Вывод результата на экран
mov eax, div
call sprint
mov eax, edi
call iprintLF
call quit

```

Рис. 4.1: Текст программы

Создаю исполняемый файл и проверяю его работу для значений $x_1=1$ и $x_2=5$ из таблицы 6.3.(рис. 4.2)


```
[vkvoinov@fedora lab06]$ nasm -f elf 18.asm
[vkvoinov@fedora lab06]$ ld -m elf_i386 -o 18 18.o
[vkvoinov@fedora lab06]$ ./18
f(x)=3(x+10)-20
Введите x:
1
Результат: 13
[vkvoinov@fedora lab06]$ ./18
f(x)=3(x+10)-20
Введите x:
5
Результат: 25
[vkvoinov@fedora lab06]$
```

Рис. 4.2: Вывод программы

Программа вывела выражение для вычисления, вывела запрос на ввод значения x , вычислила заданное выражение в зависимости от введенного x , вывела результат вычислений.

5 Вывод

Выполняя эту лабораторную работу я освоил арифметических инструкций языка ассемблера NASM.