

Отчёт по лабораторной работе №6

НПМбв-02-21

Гугульян Ксения Александровна

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	8
4	Выводы	15
	Список литературы	16

Список иллюстраций

3.1	Создание файла	8
3.2	Введение в файл текст программы	8
3.3	Создание исполняемого файла	9
3.4	Исправление текста	9
3.5	Создание исп. файла	9
3.6	Создание файла	9
3.7	Введение в файл текст	10
3.8	Создание исп. файла	10
3.9	Изменение символов	10
3.10	Создание исп. файла	10
3.11	Замена функций	11
3.12	Создание файла	11
3.13	Введение программы	11
3.14	Создание исп. файла	12
3.15	Создание исп. файла	12
3.16	Введение текста в файл	13
3.17	Создание исп. файла	13

Список таблиц

1 Цель работы

Освоение арифметических инструкций языка ассамблера NASM.

2 Задание

1. Создайте каталог для программ лабораторной работы № 6, перейдите в него и создайте файл lab6-1.asm.
2. Введите в файл lab6-1.asm текст программы из листинга 6.1. Создайте исполняемый файл и запустите его.
3. Далее изменим текст программы и вместо символов, запишем в регистры числа. Создайте исполняемый файл и запустите его.
4. Создайте файл lab6-2.asm в каталоге ~/work/arch-pc/lab06 и введите в него текст программы из листинга 6.2. Создайте исполняемый файл и запустите его.
5. Аналогично предыдущему примеру изменим символы на числа. Создайте исполняемый файл и запустите его. Какой результат будет получен при исполнении программы? Замените функцию `iprintln` на `iprint`. Создайте исполняемый файл и запустите его. Чем отличается вывод функций `iprintln` и `iprint`?
6. Создайте файл lab6-3.asm в каталоге ~/work/arch-pc/lab06. Внимательно изучите текст программы из листинга 6.3 и введите в lab6-3.asm. Создайте исполняемый файл и запустите его. Измените текст программы для вычисления выражения $\text{X}(\text{X}) = (4 \times 6 + 2)/5$. Создайте исполняемый файл и проверьте его работу.

7. Создайте файл `variant.asm` в каталоге `~/work/arch-рс/lab06`. Внимательно изучите текст программы из листинга 6.4 и введите в файл `variant.asm`. Создайте исполняемый файл и запустите его. Проверьте результат работы программы вычислив номер варианта аналитически.

3 Выполнение лабораторной работы

1. Создаём каталог для программ лабораторной работы № 6, перейдём в него и создаём файл lab6-1.asm (рис. 3.1).

```
kaguguljyan@dk6n52 ~ $ mkdir ~/work/arch-pc/lab06
kaguguljyan@dk6n52 ~ $ cd ~/work/arch-pc/lab06
kaguguljyan@dk6n52 ~/work/arch-pc/lab06 $ touch lab6-1.asm
kaguguljyan@dk6n52 ~/work/arch-pc/lab06 $
```

Рис. 3.1: Создание файла

2. Введём в файл lab6-1.asm текст программы из листинга 6.1 (рис. 3.2).

```
lab6-1.asm [-----] 0 L:[ 1+13 14/ 14] *(178
#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. 3.2: Введение в файл текст программы

Создаём исполняемый файл и запускаем его (рис. 3.3).


```
kaguguljyan@dk6n55 ~/work/arch-pc/lab06 $ nasm -f elf lab6-1.asm
kaguguljyan@dk6n55 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-1 lab6-1.o
kaguguljyan@dk6n55 ~/work/arch-pc/lab06 $ ./lab6-1
j
kaguguljyan@dk6n55 ~/work/arch-pc/lab06 $
```

Рис. 3.3: Создание исполняемого файла

3. Исправим текст программы (Листинг 6.1) (рис. 3.4).

```
lab6-1.asm [----] 12 L: [ 1+ 8 9/ 14] *(119 / 174b) 0010 0x00A
%include "inc_nasm.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 sprintf
call quit
```

Рис. 3.4: Исправление текста

Создаём исполняемый файл и запустите его (рис. 3.5).

```
kaguguljyan@dk6n55 ~/work/arch-pc/lab06 $ nasm -f elf lab6-1.asm
kaguguljyan@dk6n55 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-1 lab6-1.o
kaguguljyan@dk6n55 ~/work/arch-pc/lab06 $ ./lab6-1

kaguguljyan@dk6n55 ~/work/arch-pc/lab06 $
```

Рис. 3.5: Создание исп. файла

Отображается ли 10 при выводе на экран? Нет.

4. Создаём файл lab6-2.asm в каталоге ~/work/arch-pc/lab06 (рис. 3.6).

```
kaguguljyan@dk6n55 ~/work/arch-pc/lab06 $ touch ~/work/arch-pc/lab06/lab6-2.asm
```

Рис. 3.6: Создание файла

Введём в него текст программы из листинга 6.2 (рис. 3.7).

```
lab6-2.asm [----] 0 L:[ 1+ 9 10/ 10] *(121 / 121b) <EOF>
#include "in_out.asm"
SECTION .text
GLOBAL _start
_start:
mov eax, '1'
mov ebx, '4'
add eax, ebx
call iprintLF
call quit
```

Рис. 3.7: Введение в файл текст

Создаём исполняемый файл и запускаем его (рис. 3.8).

```
kaguguljyan@dk6n55 ~/work/arch-pc/lab06 $ nasm -f elf lab6-2.asm
kaguguljyan@dk6n55 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-2 lab6-2.o
kaguguljyan@dk6n55 ~/work/arch-pc/lab06 $ ./lab6-2
106
kaguguljyan@dk6n55 ~/work/arch-pc/lab06 $
```

Рис. 3.8: Создание исп. файла

5. Аналогично предыдущему примеру изменим символы на числа (рис. 3.9).

```
lab6-2.asm [----] 0 L:[ 1+ 9 10/ 10] *(117 / 117b) <EOF>
#include "in_out.asm"
SECTION .text
GLOBAL _start
_start:
mov eax, 6
mov ebx, 4
add eax, ebx
call iprintLF
call quit
```

Рис. 3.9: Изменение символов

Создаём исполняемый файл и запускаем его (рис. 3.10).

```
kaguguljyan@dk6n55 ~/work/arch-pc/lab06 $ nasm -f elf lab6-2.asm
kaguguljyan@dk6n55 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-2 lab6-2.o
kaguguljyan@dk6n55 ~/work/arch-pc/lab06 $ ./lab6-2
10
kaguguljyan@dk6n55 ~/work/arch-pc/lab06 $
```

Рис. 3.10: Создание исп. файла

Какой результат будет получен при исполнении программы? 10.

Заменим функцию `iprintLF` на `iprint`. Создайте исполняемый файл и запустите его. `iprintLF` и `iprint` отличаются тем, что `iprintLF` оставляет пробел, а `iprint` нет (рис. 3.11).

```
kaguguljyan@dk6n55 ~/work/arch-pc/lab06 $ nasm -f elf lab6-2.asm
kaguguljyan@dk6n55 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-2 lab6-2.o
kaguguljyan@dk6n55 ~/work/arch-pc/lab06 $ ./lab6-2
10kaguguljyan@dk6n55 ~/work/arch-pc/lab06 $
```

Рис. 3.11: Замена функций

6. Создаём файл `lab6-3.asm` в каталоге `~/work/arch-pc/lab06` (рис. 3.12).

```
10kaguguljyan@dk6n55 ~/work/arch-pc/lab06 $ touch ~/work/arch-pc/lab06/lab6-3.asm
kaguguljyan@dk6n55 ~/work/arch-pc/lab06 $
```

Рис. 3.12: Создание файла

Внимательно изучим текст программы из листинга 6.3 и введём в `lab6-3.asm` (рис. 3.13).

```
lab6-3.asm [----] 0 L: [ 1+24 25/ 26] *(344 / 345b) 0010 0x00A
%include "in_out.asm"
SECTION .data
div: DB "Peaynstar" ,0
rem: DB "Ostatok ot deleniya: ",0
SECTION .text
GLOBAL _start
_start:
mov eax,5
mov ebx,2
mul ebx
add eax,3
xor edx,edx
mov ebx,3
div ebx
mov edi,eax
mov eax,div
call sprint
mov eax,edi
call iprintLF
mov eax,rem
call sprint
mov eax,edx
call iprintLF
call quit
```

Рис. 3.13: Введение программы

Создаём исполняемый файл и запускаем его (рис. 3.14).

```
kaguguljyan@dk6n55 ~/work/arch-pc/lab06 $ nasm -f elf lab6-3.asm
kaguguljyan@dk6n55 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-3 lab6-3.o
kaguguljyan@dk6n55 ~/work/arch-pc/lab06 $ ./lab6-3
Результат: 4
Остаток от деления: 1
kaguguljyan@dk6n55 ~/work/arch-pc/lab06 $
```

Рис. 3.14: Создание исп. файла

Изменим текст программы для вычисления выражения $\boxtimes(\boxtimes) = (4 \boxtimes 6 + 2)/5$.

Создаём исполняемый файл и проверяем его работу (рис. 3.15).

```
kaguguljyan@dk6n55 ~/work/arch-pc/lab06 $ nasm -f elf lab6-3.asm
kaguguljyan@dk6n55 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-3 lab6-3.o
kaguguljyan@dk6n55 ~/work/arch-pc/lab06 $ ./lab6-3
Результат: 5
Остаток от деления: 1
kaguguljyan@dk6n55 ~/work/arch-pc/lab06 $
```

Рис. 3.15: Создание исп. файла

7. Создаём файл `variant.asm` в каталоге `~/work/arch-pc/lab06`: `touch ~/work/arch-pc/lab06/variant.asm`. Внимательно изучим текст программы из листинга 6.4 и введём в файл `variant.asm` (рис. 3.16).

```

variant.asm [----] 9 L: [ 1+24 25/ 25] *(384 / 384b) <EOF>
#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
inc edx
mov eax, rem
call sprint
mov eax, edx
call iprintLF
call quit

```

Рис. 3.16: Введение текста в файл

Создаём исполняемый файл и запускаем его (рис. 3.17).

```

kaguguljyan@dk6n55 ~/work/arch-pc/lab06 $ nasm -f elf variant.asm
kaguguljyan@dk6n55 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o variant variant.o
kaguguljyan@dk6n55 ~/work/arch-pc/lab06 $ ./variant
Введите № студенческого билета:
1032217836
Ваш вариант: 17
kaguguljyan@dk6n55 ~/work/arch-pc/lab06 $

```

Рис. 3.17: Создание исп. файла

Вопросы:

1. `xor edx,edx mov ebx,20 div ebx inc edx mov eax,rem call sprint mov eax,edx call iprintLF call quit`
2. Используются для чтения строки с клавиатуры и сохранения её в памяти.
3. Используется для вызова функции `atoi`, которая преобразует строку в целое число.
4. `xor edx,edx mov ebx,20 div ebx inc edx`

5. Результат деления записывается в регистр `eax`, а остаток от деления записывается в регистр `edx`.
6. Используется для увеличения значения в регистре `edx` на 1.
7. `mov eax,rem call sprint mov eax,edx call iprintLF call quit`

4 Выводы

В ходе решения лабораторной работы я освоила арифметические инструкции языка ассамблера NASM.

Список литературы