

Отчет по лабораторной работе №6

Дисциплина: архитектура компьютера

Ардеев Никита Евгеньевич НММбд-01-23

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Ответы на вопросы	13
4	Самостоятельная работа	14
5	Выводы	16

Список иллюстраций

2.1	Создание каталога и файла	6
2.2	Текст программы	7
2.3	Работа программы	7
2.4	Замена символов на числа	8
2.5	Работа программы	8
2.6	Создание файла	8
2.7	Работа программы	9
2.8	Текст программы	9
2.9	Работа программы	9
2.10	Вывод при команде <code>iprint</code>	10
2.11	Работа программы	10
2.12	Текст программы	11
2.13	Работа программы	11
2.14	Создание файла	11
2.15	Работа программы	12
4.1	Работа программы	14
4.2	Текст программы	15

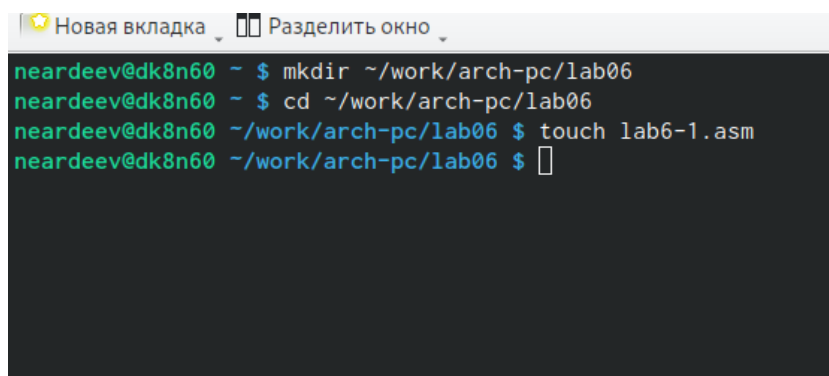
Список таблиц

1 Цель работы

Освоение арифметических инструкций языка ассемблера NASM.

2 Выполнение лабораторной работы

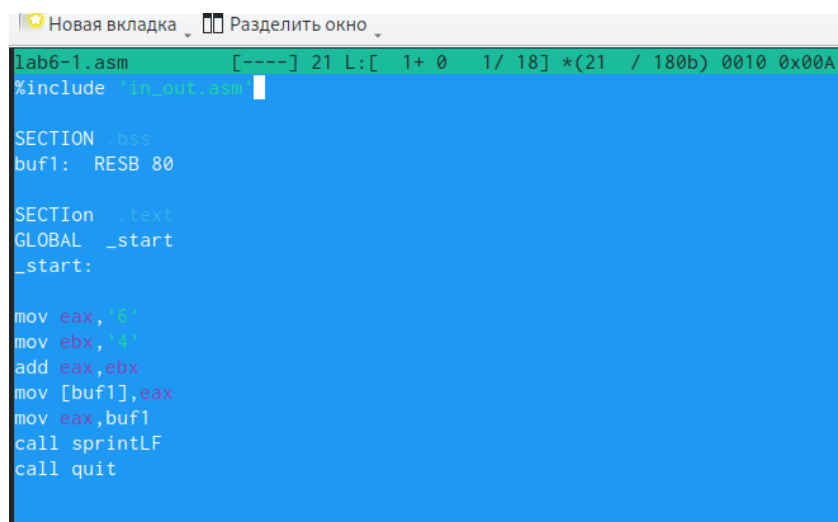
Создал каталог для программ лабораторной работы №6, перейдя в него создал файл lab6-1.asm(рис. 2.1).



```
neardeev@dk8n60 ~ $ mkdir ~/work/arch-pc/lab06
neardeev@dk8n60 ~ $ cd ~/work/arch-pc/lab06
neardeev@dk8n60 ~/work/arch-pc/lab06 $ touch lab6-1.asm
neardeev@dk8n60 ~/work/arch-pc/lab06 $
```

Рис. 2.1: Создание каталога и файла

Ввел в файл lab6-1.asm текст программы из листинга(рис. 2.2).



```
lab6-1.asm [----] 21 L: [ 1+ 0 1/ 18] *(21 / 180b) 0010 0x00A
#include "in_out.asm"

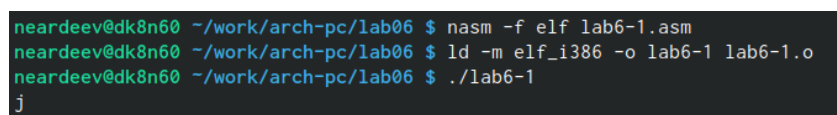
SECTION .bss
buf1: RESB 80

SECTION .text
GLOBAL _start
_start:

mov eax, '6'
mov ebx, '4'
add eax, ebx
mov [buf1], eax
mov eax, buf1
call sprintf
call quit
```

Рис. 2.2: Текст программы

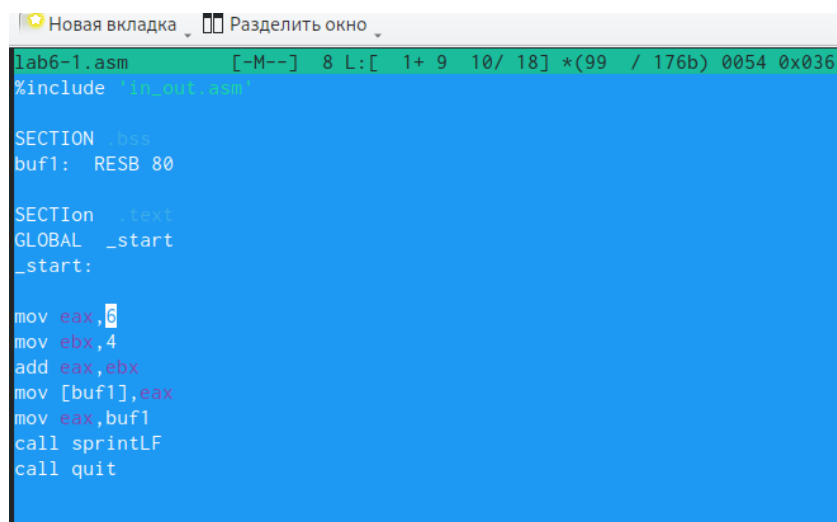
Создал исполняемый файл и запустил его(рис. 2.3).



```
neardeev@dk8n60 ~/work/arch-pc/lab06 $ nasm -f elf lab6-1.asm
neardeev@dk8n60 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-1 lab6-1.o
neardeev@dk8n60 ~/work/arch-pc/lab06 $ ./lab6-1
j
```

Рис. 2.3: Работа программы

Изменил текст программы, записав числа в регистр вместо символов(рис. 2.4).



```
lab6-1.asm [-M--] 8 L: [ 1+ 9 10/ 18] *(99 / 176b) 0054 0x036
#include "in_out.asm"

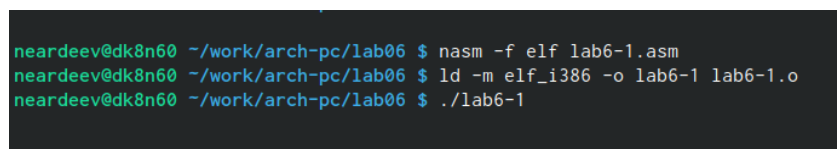
SECTION .bss
buf1: RESB 80

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: Замена символов на числа

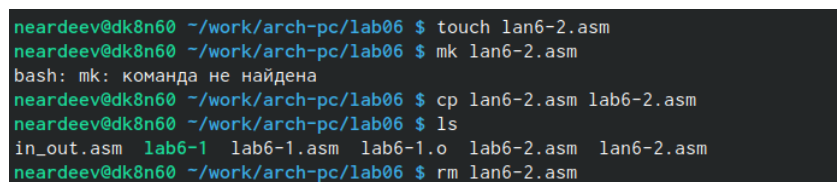
Создал исполняемый файл и запустил его. По таблице ASCII код 10 соответствует символу Line FEED (то есть переход на новую строку, данный символ не отображается.) (рис. 2.5).



```
neardeev@dk8n60 ~/work/arch-pc/lab06 $ nasm -f elf lab6-1.asm
neardeev@dk8n60 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-1 lab6-1.o
neardeev@dk8n60 ~/work/arch-pc/lab06 $ ./lab6-1
```

Рис. 2.5: Работа программы

Создал файл lab6-2.asm в каталоге ~/work/arch-pc/lab06 и ввел в него текст программы из листинга (рис. 2.6).



```
neardeev@dk8n60 ~/work/arch-pc/lab06 $ touch lan6-2.asm
neardeev@dk8n60 ~/work/arch-pc/lab06 $ mk lan6-2.asm
bash: mk: команда не найдена
neardeev@dk8n60 ~/work/arch-pc/lab06 $ cp lan6-2.asm lab6-2.asm
neardeev@dk8n60 ~/work/arch-pc/lab06 $ ls
in_out.asm lab6-1 lab6-1.asm lab6-1.o lab6-2.asm lan6-2.asm
neardeev@dk8n60 ~/work/arch-pc/lab06 $ rm lan6-2.asm
```

Рис. 2.6: Создание файла

Создал исполняемый файл и запустил его (рис. 2.7).


```

neardeev@dk8n60 ~/work/arch-pc/lab06 $ nasm -f elf lab6-2.asm
neardeev@dk8n60 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-2 lab6-2.o
ld: не распознан режим эмуляции: elf_i386
Поддерживаемые эмуляции: elf_x86_64 elf32_x86_64 elf_i386 elf_iamcu
neardeev@dk8n60 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-2 lab6-2.o
neardeev@dk8n60 ~/work/arch-pc/lab06 $ ./lab6-2
106
neardeev@dk8n60 ~/work/arch-pc/lab06 $

```

Рис. 2.7: Работа программы

Снова заменил символы на числа(рис. 2.8).

```

lab6-2.asm      [-M--]  8 L:[  1+ 6   7/ 11] *(77 / 115b) 0052 0x034
#include "io_out.asm"

SECTION .text
GLOBAL _start
_start:
mov eax,6
mov ebx,4
add eax,ebx
call iprintLF
call quit

```

Рис. 2.8: Текст программы

Создал исполняемый файл и запустил его. В результате вышла сумма чисел: 10(рис. 2.9).

```

neardeev@dk8n60 ~/work/arch-pc/lab06 $ mc

neardeev@dk8n60 ~/work/arch-pc/lab06 $ nasm -f elf lab6-2.asm
neardeev@dk8n60 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-2 lab6-2.o
neardeev@dk8n60 ~/work/arch-pc/lab06 $ ./lab6-2
10
neardeev@dk8n60 ~/work/arch-pc/lab06 $

```

Рис. 2.9: Работа программы

Заменял функцию iprintLF на . Создай исполняемый файл и запустил его. iprintLF выводит результат в отдельной строке, а iprint выводит на той же строке, где предлагается ввод новой команды.(рис. 2.10).

```

neardeev@dk8n60 ~/work/arch-pc/lab06 $ nasm -f elf lab6-2.asm
neardeev@dk8n60 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-2 lab6-2.o
neardeev@dk8n60 ~/work/arch-pc/lab06 $ ./lab6-2
10neardeev@dk8n60 ~/work/arch-pc/lab06 $ █

```

Рис. 2.10: Вывод при команде `iprint`

Создал файл `lab6-3.asm` в каталоге `~/work/arch-pc/lab06`, ввел в нем текст программы из листинга, затем создал исполняемый файл и запустил его (рис. 2.11).

```

neardeev@dk8n60 ~/work/arch-pc/lab06 $ nasm -f lab6-3.asm
nasm: fatal: unrecognized output format 'lab6-3.asm' - use -hf for a list
Type nasm -h for help.
neardeev@dk8n60 ~/work/arch-pc/lab06 $ nasm -f elf lab6-3.asm
neardeev@dk8n60 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-3 lab6-3.o
neardeev@dk8n60 ~/work/arch-pc/lab06 $ ./lab6-3
Результат: 4
Остаток от деления: 1
neardeev@dk8n60 ~/work/arch-pc/lab06 $ █

```

lab06: bash — Konsole

Рис. 2.11: Работа программы

Изменил текст программы для вычисления выражения $\boxtimes(\boxtimes) = (4 \boxtimes 6 + 2)/5$ (рис. 2.12).

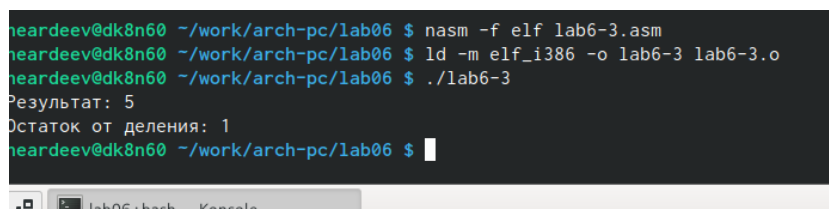


```
lab6-3.asm [----] 9 L: [ 1+13 14/ 26] *(214 / 345b) 0010 0x00A
#include "in_out.asm"
SECTION .data
div: DB "Результат: ",0
rem: DB "Остаток от деления: ",0
SECTION .text
GLOBAL _start
_start:

mov eax,4
mov ebx,6
mul ebx
add eax,2
xor edx,edx
mov ebx,5
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. 2.12: Текст программы

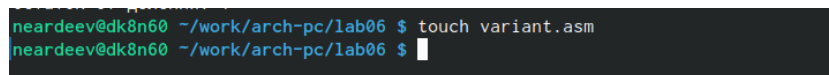
Создал исполняемый файл и запустил его(рис. 2.13).



```
neardeev@dk8n60 ~/work/arch-pc/lab06 $ nasm -f elf lab6-3.asm
neardeev@dk8n60 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-3 lab6-3.o
neardeev@dk8n60 ~/work/arch-pc/lab06 $ ./lab6-3
Результат: 5
Остаток от деления: 1
neardeev@dk8n60 ~/work/arch-pc/lab06 $
```

Рис. 2.13: Работа программы

Создал файл variant.asm в каталоге ~/work/arch-pc/lab06(рис. 2.14).



```
neardeev@dk8n60 ~/work/arch-pc/lab06 $ touch variant.asm
neardeev@dk8n60 ~/work/arch-pc/lab06 $
```

Рис. 2.14: Создание файла

Ввел в нем текст программы из листинга,создал исполняемый файл и запустил его(рис. 2.15).

```
neardeev@dk8n60 ~/work/arch-pc/lab06 $ nasm -f elf variant.asm
neardeev@dk8n60 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o variant variant.o
ld: невозможно найти variant.o: Нет такого файла или каталога
neardeev@dk8n60 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o variant variant.o
neardeev@dk8n60 ~/work/arch-pc/lab06 $ ./variant
Введите № студенческого билета:
1132236090
Ваш вариант: 11
neardeev@dk8n60 ~/work/arch-pc/lab06 $
```

Рис. 2.15: Работа программы

3 Ответы на вопросы

1. `mov eax,rem call sprint.`
2. Сначала определяет адрес значения “х” в “есх”, а затем определяет длину вводимой строки.
3. Для преобразования ACSII кода в число
4. `xor edx,edx mov ebx,20 div ebx inc edx`
5. `edx`
6. команда `inc ebx` увеличивает значение регистра `edx` на 1
7. `mov eax,edx call iprintLF`

4 Самостоятельная работа

Я написал программу для вычисления уравнения $y = 10(x + 1) - 10$, которая в зависимости от введенного x вычисляет его значение. Для проверки использовал $x_1=1, x_2=7$ (рис. 4.1).

```
neardeev@dk8n60 ~/work/arch-pc/lab06 $ nasm -f elf samrab.asm
neardeev@dk8n60 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o samrab samrab.o
neardeev@dk8n60 ~/work/arch-pc/lab06 $ ./samrab
Введите переменную:
1
Результат: 10
neardeev@dk8n60 ~/work/arch-pc/lab06 $ ./samrab
Введите переменную:
7
Результат: 70
neardeev@dk8n60 ~/work/arch-pc/lab06 $
```

Рис. 4.1: Работа программы

```

SamirAB.asm [~] 7 L: [ 1+23 247 34] x(294 / 4040) 0010 0x00A
%include "in_out.asm"
SECTION .data
rem: DB 'Результат: ',0
msg: DB 'Введите переменную: ',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

add eax, 1
mov ebx, 10
mul ebx
xor edx, edx
mov ebx, -10
add eax, ebx
mov edi, eax

mov eax, rem
call sprintf
mov eax, edi
call iprintLF
call quit

```

Рис. 4.2: Текст программы

5 Выводы

В ходе выполнения лабораторной работы я ознакомился с арифметическими действиями в NASM, научился составлять программу, решающую уравнение с одной переменной.