

# **Отчёт по лабораторной работе № 6**

**Дисциплина: Архитектура компьютера**

Вакутайпа Милдред

# Содержание

|          |  |           |
|----------|--|-----------|
| <b>1</b> | <b>Цель работы</b>                       | <b>4</b>  |
| <b>2</b> | <b>Задание</b>                           | <b>5</b>  |
| <b>3</b> | <b>Выполнение лабораторной работы</b>    | <b>6</b>  |
| <b>4</b> | <b>Выполнение самостоятельной работы</b> | <b>17</b> |
| <b>5</b> | <b>Выводы</b>                            | <b>21</b> |
|          | <b>Список литературы</b>                 | <b>22</b> |

# Список иллюстраций

|      |        |    |
|------|--------|----|
| 3.1  | Рис 1  | 6  |
| 3.2  | Рис 2  | 6  |
| 3.3  | Рис 3  | 7  |
| 3.4  | Рис 4  | 7  |
| 3.5  | Рис 5  | 7  |
| 3.6  | Рис 6  | 8  |
| 3.7  | Рис 7  | 8  |
| 3.8  | Рис 8  | 8  |
| 3.9  | Рис 9  | 8  |
| 3.10 | Рис 10 | 9  |
| 3.11 | Рис 11 | 9  |
| 3.12 | Рис 12 | 9  |
| 3.13 | Рис 13 | 10 |
| 3.14 | Рис 14 | 10 |
| 3.15 | Рис 15 | 10 |
| 3.16 | Рис 16 | 11 |
| 3.17 | Рис 17 | 12 |
| 3.18 | Рис 18 | 13 |
| 3.19 | Рис 19 | 14 |
| 3.20 | Рис 20 | 14 |
| 3.21 | Рис 21 | 14 |
| 3.22 | Рис 22 | 15 |
| 4.1  | Рис 23 | 17 |
| 4.2  | Рис 24 | 18 |
| 4.3  | Рис 25 | 19 |
| 4.4  | Рис 26 | 19 |

# 1 Цель работы

Освоение арифметических инструкций языка ассемблера NASM.

## 2 Задание

1. Символьные и численные данные в NASM
2. Выполнение арифметических операций в NASM
3. Ответы на вопросы по программе

## 3 Выполнение лабораторной работы

### 1. Символьные и численные данные в NASM

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

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

Рис. 3.1: Рис 1

Копирую в текущий каталог файл in\_out.asm, потому что он будет использоваться в других программах:

```
mwakutaipa@mwakutaipa:~/work/arch-pc/lab06$ cp ~/Downloads/in_out.asm in_out.asm
mwakutaipa@mwakutaipa:~/work/arch-pc/lab06$ ls
in_out.asm  lab6-1.asm
```

Рис. 3.2: Рис 2

Открываю созданный файл lab6-1.asm, вставляю в него программу вывода значения регистра eax:

```

GNU nano 6.2 /home/mwakutaipa/work/arch-pc/lab06/lab6-1.asm *
#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. 3.3: Рис 3

Создаю исполняемый файл программы и запускаю его. Программа выводит символ 'j', потому что это сумма кодов символов 6 и 4 по системе ASCII:

```

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

```

Рис. 3.4: Рис 4

Изменяю текст программы и вместо символов, пишу в регистры числа:

```

GNU nano 6.2 /home/mwakutaipa/work/arch-pc/lab06/lab6-1.asm *
#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. 3.5: Рис 5

Создаю исполняемый файл программы и запускаю его. Теперь она выводит новую строку (код символа 10):

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

Рис. 3.6: Рис 6

Создаю новый файл lab6-2.asm:

```
Hint: Want your plain shell? Press C-o, and get back to MC with C-o again.
mwakutaipa@mwakutaipa:~/work/arch-pc/lab06$ touch lab6-2.asm
1Help 2Menu 3View 4Edit 5Copy 6RenMov 7Mkdir 8Delete 9PullDn10Quit
```

Рис. 3.7: Рис 7

Открываю созданный файл lab6-2.asm, вставляю в него программу вывода значения регистра eax:

```
GNU nano 6.2 /home/mwakutaipa/work/arch-pc/lab06/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. 3.8: Рис 8

Создаю исполняемый файл программы и запускаю его. Она выводит сумму кодов символ 6(54) и символ 4(52):

```
mwakutaipa@mwakutaipa:~/work/arch-pc/lab06$ nasm -f elf lab6-2.asm
```

Рис. 3.9: Рис 9



```
mwakutaipa@mwakutaipa:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-2 lab6-2.o
mwakutaipa@mwakutaipa:~/work/arch-pc/lab06$ ./lab6-2
106
```

Рис. 3.10: Рис 10

Изменяю текст программы и вместо символов, пишу в регистры числа:

```
GNU nano 6.2 /home/mwakutaipa/work/arch-pc/lab06/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. 3.11: Рис 11

Создаю исполняемый файл программы и запускаю его. Теперь она выводит сумму чисел 6 и 4:

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

Рис. 3.12: Рис 12

Изменяю iprintLF в iprint:

```
GNU nano 6.2 /home/mwakutaipa/work/arch-pc/lab06/lab6-2.asm
%include 'in_out.asm'

SECTION .text
GLOBAL _start

_start:
mov eax,6
mov ebx,4
add eax,ebx
call iprint
call quit
```

Рис. 3.13: Рис 13

Создаю исполняемый файл и запускаю его:

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

Рис. 3.14: Рис 14

## 2. Выполнение арифметических операций в NASM

Создаю новый файл lab6-3.asm:

```
Hint: Completion: use M-Tab (or Esc+Tab). Type it twice to get a list.
mwakutaipa@mwakutaipa:~/work/arch-pc/lab06$ touch lab6-3.asm
1Help 2Menu 3View 4Edit 5Copy 6RenMov 7Mkdir 8Delete 9PullDn10Quit
```

Рис. 3.15: Рис 15

Вставляю в него программу для вычисления выражения  $f(x) = (5 * 2 + 3)/3$  :

```
GNU nano 6.2
#include 'in_out.asm'

SECTION .data

div: DB 'Результат: ', 0
rem: DB 'Остаток от деления: ', 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. 3.16: Рис 16

Создаю исполняемый файл и запускаю его:

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

Рис. 3.17: Рис 17

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

```
GNU nano 6.2
#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. 3.18: Рис 18

Создаю исполняемый файл и запускаю его:

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

Рис. 3.19: Рис 19

Создаю файл variant.asm:

```
Hint: Tired of these messages? Turn them off from the Options/Layout menu.
mwakutaipa@mwakutaipa:~/work/arch-pc/lab06$ touch variant.asm
1Help 2Menu 3View 4Edit
```

Рис. 3.20: Рис 20

Ввожу в файл текст программы для вычисления варианта задания по номеру студенческого билета:

```
GNU nano 6.2
#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 sprintLF

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. 3.21: Рис 21

Создаю исполняемый файл и запускаю его. Ввожу номер своего студенческого билета, программа выводит, вариант 10:

```
mwakutaipa@mwakutaipa:~/work/arch-pc/lab06$ nasm -f elf variant.asm
mwakutaipa@mwakutaipa:~/work/arch-pc/lab06$ ld -m elf_i386 -o variant variant.o
mwakutaipa@mwakutaipa:~/work/arch-pc/lab06$ ./variant
Введите № студенческого билета:
1032239009
Ваш вариант: 10
```

Рис. 3.22: Рис 22

### 3. Ответы на вопросы по программе

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

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

2. `mov ecx, x` используется, чтобы положить адрес вводимой строки `x` в регистр. `ecx mov edx, 80` – запись в регистр `edx` длины вводимой строки `call sread` – вызов подпрограммы из внешнего файла, обеспечивающей ввод общения 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
mov ebx,20
div ebx
inc edx
```

5. Остаток от деления записывается в регистр `edx`.

6. Инструкция `inc edx` используется для увеличения значения регистра `edx` на 1.

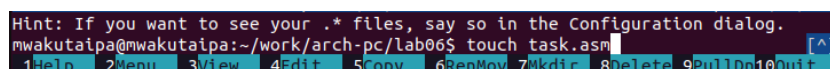
7. За вывод на экран результатов вычислений отвечают строки:

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



## 4 Выполнение самостоятельной работы

Создаю файл task.asm:



```
Hint: If you want to see your .* files, say so in the Configuration dialog.
mwakutaipa@mwakutaipa:~/work/arch-pc/lab06$ touch task.asm
```

Рис. 4.1: Рис 23

В него пишу программу для вычисления выражения  $f(x)=5(x + 18) - 28$ . Она берет входное значение " x ", прибавляет его к 18, умножает результат сложения на 5 а потом вычитает 28:

```

GNU nano 6.2
%include 'in_out.asm'

SECTION .data

msg: DB 'Введите значение x: ', 0
rem: DB 'Результат: ', 0

SECTION .bss
x: RESB 80

SECTION .text
GLOBAL _start
_start:
mov eax,msg
call sprint

mov ecx,x
mov edx,80
call sread

mov eax,x
call atoi

add eax,18
mov ebx,5
mul ebx
add eax,-28
mov edi,eax

mov eax,rem
call sprint

mov eax,edi
call iprint

call quit

```

Рис. 4.2: Рис 24

Создаю исполняемый файл и запускаю его. При  $x = 2$ , она выводит 72. При  $x = 3$ , она выводит 77:

```
mwakutaipa@mwakutaipa:~/work/arch-pc/lab06$ nasm -f elf task.asm
```

Рис. 4.3: Рис 25

```
mwakutaipa@mwakutaipa:~/work/arch-pc/lab06$ ld -m elf_i386 -o task task.o
mwakutaipa@mwakutaipa:~/work/arch-pc/lab06$ ./task
Введите значение x: 2
Результат: 72
mwakutaipa@mwakutaipa:~/work/arch-pc/lab06$ ./task
Введите значение x: 3
Результат: 77
```

Рис. 4.4: Рис 26

Код Программы:

```
%include 'in_out.asm'
```

```
SECTION .data
```

```
msg: DB 'Введите значение переменной x: ',0
```

```
rem: DB 'Результат: ',0
```

```
SECTION .bss
```

```
x: RESB 80
```

```
SECTION .text
```

```
GLOBAL _start
```

```
_start:
```

```
mov eax, msg
```

```
call sprint
```

```
mov ecx, x
```

```
mov edx, 80
```

```
call sread
```

```
mov eax,x
```

```
call atoi
```

```
add eax,18
```

```
mov ebx,5
```

```
mul ebx
```

```
add eax,-28
```

```
mov edi,eax
```

```
mov eax,rem
```

```
call sprint
```

```
mov eax,edi
```

```
call iprint
```

```
call quit
```

## 5 Выводы

При выполнении лабораторной работы я освоила арифметические инструкции языка ассемблера NASM.

# Список литературы

Архитектура ЭВМ :::