

# **Отчёт по лабораторной работе №7**

**Дисциплина: Архитектура компьютера**

**Ищенко Ирина Олеговна**

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Выполнение лабораторной работы</b>	<b>6</b>
<b>3</b>	<b>Выполнение заданий для самостоятельной работы</b>	<b>14</b>
<b>4</b>	<b>Выводы</b>	<b>16</b>

## Список иллюстраций

2.1	Создание каталога и файла . . . . .	6
2.2	Первый запуск программы вывода значения регистра eax . . . . .	7
2.3	Второй запуск программы . . . . .	7
2.4	Первый запуск второй программы . . . . .	8
2.5	Второй запуск второй программы . . . . .	8
2.6	Третий запуск второй программы . . . . .	8
2.7	Запуск программы вычисления выражения $(5 * 2 + 3)/3$ . . . . .	9
2.8	Запуск программы вычисления выражения $(4 * 6 + 2)/5$ . . . . .	11
2.9	Программа вычисления вычисления варианта задания по номеру студенческого билета . . . . .	12
3.1	Программа вычисления выражения $f(x)=5*(x+18)-28$ . . . . .	15

## Список таблиц

# 1 Цель работы

Освоение арифметических инструкций языка ассемблера NASM.

## 2 Выполнение лабораторной работы

Создадим каталог для программ лабораторной работы № 7, переходим в него и создаем файл lab7-1.asm (рис. 2.1).

```
[ioithenko@fedora ~]$ mkdir ~/work/arch-pc/lab07  
[ioithenko@fedora ~]$ cd ~/work/arch-pc/lab07  
[ioithenko@fedora lab07]$ touch lab7-1.asm
```

Рис. 2.1: Создание каталога и файла

Введем в файл lab7-1.asm текст программы из листинга 1, создадим исполняемый файл и запустим его (рис. 2.2). Листинг 1:

```
%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
```

```
[ioithenko@fedora lab07]$ nasm -f elf lab7-1.asm
[ioithenko@fedora lab07]$ ld -m elf_i386 -o lab7-1 lab7-1.o
[ioithenko@fedora lab07]$ ./lab7-1
j
[ioithenko@fedora lab07]$
```

Рис. 2.2: Первый запуск программы вывода значения регистра eax

Далее изменим текст программы и вместо символов запишем в регистры числа. Создадим исполняемый файл и запустим его (рис. 2.3). Согласно таблице ASCII, коду 10 соответствует символ LF. При выходе на экран этот символ не отображается, так как является переносом на новую строку.

```
[ioithenko@fedora lab07]$ mc
[ioithenko@fedora lab07]$ nasm -f elf lab7-1.asm
[ioithenko@fedora lab07]$ ld -m elf_i386 -o lab7-1 lab7-1.o
[ioithenko@fedora lab07]$ ./lab7-1

[ioithenko@fedora lab07]$ touch ~/work/arch-pc/lab07/lab7-2.asm
```

Рис. 2.3: Второй запуск программы

Создадим файл lab7-2.asm в каталоге ~/work/arch-pc/lab07 и введем в него текст программы из листинга 2. Создадим исполняемый файл и запустим его (рис. 2.4).  
Листинг 2:

```
%include 'in_out.asm'

SECTION .text
GLOBAL _start

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

```
[ioithenko@fedora lab07]$ touch ~/work/arch-pc/lab07/lab7-2.asm
[ioithenko@fedora lab07]$ mc

[ioithenko@fedora lab07]$ nasm -f elf lab7-2.asm
[ioithenko@fedora lab07]$ ld -m elf_i386 -o lab7-2 lab7-2.o
[ioithenko@fedora lab07]$ ./lab7-2
106
[ioithenko@fedora lab07]$
```

Рис. 2.4: Первый запуск второй программы

Изменим текст программы и вместо символов запишем в регистры числа. Создадим исполняемый файл и запустим его (рис. 2.5). В результате программа выведет 10.

```
[ioithenko@fedora lab07]$ mc

[ioithenko@fedora lab07]$ nasm -f elf lab7-2.asm
[ioithenko@fedora lab07]$ ld -m elf_i386 -o lab7-2 lab7-2.o
[ioithenko@fedora lab07]$ ./lab7-2
10
[ioithenko@fedora lab07]$
```

Рис. 2.5: Второй запуск второй программы

Заменяем функцию `iprintlnf` на `iprint`. Создадим исполняемый файл и запустим его (рис. 2.6). Функция `iprintlnf` выводит значение с переносом на новую строку в отличие от `iprint`.

```
[ioithenko@fedora lab07]$ mc

[ioithenko@fedora lab07]$ nasm -f elf lab7-2.asm
[ioithenko@fedora lab07]$ ld -m elf_i386 -o lab7-2 lab7-2.o
[ioithenko@fedora lab07]$ ./lab7-2
10[ioithenko@fedora lab07]$
```

Рис. 2.6: Третий запуск второй программы

Создадим файл `lab7-3.asm` в каталоге `~/work/arch-pc/lab07` и введем текст программы из листинга 3. Создадим исполняемый файл и запустим его (рис. 2.7). Листинг 3:

```
;-----
; Программа вычисления выражения
;-----
%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,5 ; EAX=5
mov ebx,2 ; EBX=2
mul ebx ; EAX=EAX*EBX
add eax,3 ; EAX=EAX+3
xor edx,edx ; обнуляем EDX для корректной работы div
mov ebx,3 ; EBX=3
div ebx ; EAX=EAX/3, 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

```

```

[ioithenko@fedora lab07]$ nasm -f elf lab7-3.asm
[ioithenko@fedora lab07]$ ld -m elf_i386 -o lab7-3 lab7-3.o
[ioithenko@fedora lab07]$ ./lab7-3
Результат: 4
Остаток от деления: 1
[ioithenko@fedora lab07]$

```

Рис. 2.7: Запуск программы вычисления выражения  $(5 * 2 + 3)/3$

Изменим текст программы для вычисления выражения  $(4 * 6 + 2)/5$ , согласно листингу 4. Создадим исполняемый файл и проверим его работу (рис. 2.8).  
Листинг 4:

```
;-----  
; Программа вычисления выражения  
;-----  
  
%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

```

```

[ioithenko@fedora lab07]$ nasm -f elf lab7-3.asm
[ioithenko@fedora lab07]$ ld -m elf_i386 -o lab7-3 lab7-3.o
[ioithenko@fedora lab07]$ ./lab7-3
Результат: 5
Остаток от деления: 1
[ioithenko@fedora lab07]$

```

Рис. 2.8: Запуск программы вычисления выражения  $(4 * 6 + 2)/5$

Создадим файл variant.asm в каталоге ~/work/arch-pc/lab07 и введем в файл variant.asm текст из листинга 5. Создадим исполняемый файл и запустим его (рис. 2.9). Листинг 5:

```

;-----
; Программа вычисления варианта
;-----

%include 'in_out.asm'

SECTION .data
msg: DB 'Введите № студенческого билета: ',0
rem: DB 'Ваш вариант: ',0

SECTION .bss
x: RESB 80

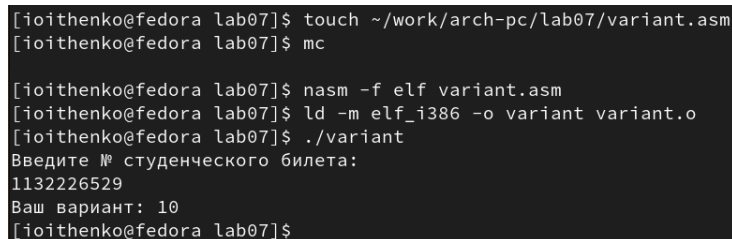
SECTION .text
GLOBAL _start
_start:
mov eax, msg
call sprintLF
mov ecx, x
mov edx, 80
call sread

```

```

mov eax,x ; вызов подпрограммы преобразования
call atoi ; ASCII кода в число, `eax=x`
xor edx,edx
mov ebx,20
div ebx
inc edx
mov eax,rem
call sprint
mov eax,edx
call iprintLF
call quit

```



```

[ioithenko@fedora lab07]$ touch ~/work/arch-pc/lab07/variant.asm
[ioithenko@fedora lab07]$ mc

[ioithenko@fedora lab07]$ nasm -f elf variant.asm
[ioithenko@fedora lab07]$ ld -m elf_i386 -o variant variant.o
[ioithenko@fedora lab07]$ ./variant
Введите № студенческого билета:
1132226529
Ваш вариант: 10
[ioithenko@fedora lab07]$

```

Рис. 2.9: Программа вычисления варианта задания по номеру студенческого билета

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

```

mov eax,rem
call sprint

```

2. Инструкция `mov esx, x` используется, чтобы положить адрес вводимой строки `x` в регистр `esx`. Инструкция `mov edx, 80` - запись в регистр `edx` длины вводимой строки. `call sread` - вызов подпрограммы ввода сообщения с клавиатуры из внешнего файла.
3. `call atoi` используется для вызова подпрограммы, которая преобразует ASCII-код символа в целое число и записывает результат в регистр `eax`.

4. За вычисления варианта отвечают строки:

```
xor edx,edx ; обнуление edx для корректной работы div
mov ebx,20 ; ebx = 20
div ebx ; eax = eax/20, edx - остаток от деления
inc edx ; edx = edx + 1
```

5. При выполнении инструкции `div ebx` остаток от деления записывается в регистр `edx`.

6. Инструкция `inc edx` увеличивает значение регистра `edx` на 1.

7. За вывод на экран результатов вычислений отвечают строки:

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

### 3 Выполнение заданий для самостоятельной работы

В соответствие с предыдущим заданием, мой номер варианта - 10ый. Напишем программу вычисления выражения  $f(x)=5*(x+18)-28$ , код программы приведен в листинге 6. Создадим исполняемый файл и проверим его работу для значений  $x_1$  и  $x_2$  (рис. 3.1). Листинг 6:

```
;-----  
; Программа вычисления выражения  
;-----  
  
%include 'in_out.asm' ; подключение внешнего файла  
  
SECTION .data  
msg: DB 'Введите значение x: ',0  
rem: DB 'Результат: ',0  
div: DB 'f(x)=5*(x+18)-28',0  
  
SECTION .bss  
x: RESB 80 ; задание переменной  
  
SECTION .text  
GLOBAL _start  
  
_start:  
; — Вычисление выражения  
  
mov eax,div ; вызов подпрограммы печати  
call sprint
```

```

mov eax, msg
call sprint
mov ecx, x
mov edx, 80
call sread
mov eax, x
call atoi
add eax, 18 ; eax = x + 18
mov ebx, 5
mul ebx ; eax = 5*(x + 18)
add eax, -28 ; eax = 5*(x + 18) - 28
mov edi, eax ; запись результата вычисления в 'edi'
; — Вывод результата на экран
mov eax, rem ; вызов подпрограммы печати
call sprint.
mov eax, edi ; вызов подпрограммы печати значения
call iprintLF ; из 'edi' в виде символов
call quit ; вызов подпрограммы завершения

```

```

[ioithenko@fedora lab07]$ mc
[ioithenko@fedora lab07]$ nasm -f elf lab7-task.asm
[ioithenko@fedora lab07]$ ld -m elf_i386 -o lab7-task lab7-task.o
[ioithenko@fedora lab07]$ ./lab7-task
f(x)=5*(x+18)-28
Введите значение x: 2
Результат: 72
[ioithenko@fedora lab07]$ ./lab7-task
f(x)=5*(x+18)-28
Введите значение x: 3
Результат: 77
[ioithenko@fedora lab07]$

```

Рис. 3.1: Программа вычисления выражения  $f(x)=5*(x+18)-28$

## 4 Выводы

В ходе выполнения лабораторной работы я освоила арифметические инструкции языка ассемблера NASM.