

# **Лабораторная работа №7**

**Дисциплина: Архитектура компьютера**

Ким Ангелина Павловна

# Содержание

|          |                                       |           |
|----------|---------------------------------------|-----------|
| <b>1</b> | <b>Цель работы</b>                    | <b>5</b>  |
| <b>2</b> | <b>Задание</b>                        | <b>6</b>  |
| <b>3</b> | <b>Выполнение лабораторной работы</b> | <b>7</b>  |
| <b>4</b> | <b>Выводы</b>                         | <b>23</b> |
|          | <b>Список литературы</b>              | <b>24</b> |

## Список иллюстраций

|      |  |    |
|------|--|----|
| 3.1  | Создание каталога . . . . .                        | 7  |
| 3.2  | Введем в файл текст программы . . . . .            | 8  |
| 3.3  | Создание исполняемого файла . . . . .              | 9  |
| 3.4  | Изменяем текст программы . . . . .                 | 10 |
| 3.5  | Создание исполняемого файла . . . . .              | 11 |
| 3.6  | Текст программы из листинга 7.2 . . . . .          | 11 |
| 3.7  | Создаем исполняемый файл . . . . .                 | 12 |
| 3.8  | Заменяем символы на числа . . . . .                | 12 |
| 3.9  | Создаем исполняемый файл и запускаем его . . . . . | 13 |
| 3.10 | Заменяем функцию . . . . .                         | 13 |
| 3.11 | Создаем исполняемый файл . . . . .                 | 14 |
| 3.12 | Текст программы . . . . .                          | 15 |
| 3.13 | Создаем исполняемый файл . . . . .                 | 16 |
| 3.14 | Изменяем текст программы . . . . .                 | 17 |
| 3.15 | Создаем исполняемый файл . . . . .                 | 18 |
| 3.16 | Текст программы из листинга 7.4 . . . . .          | 19 |
| 3.17 | Создание исполняемого файла . . . . .              | 20 |
| 3.18 | Текст программы вариант-20 . . . . .               | 21 |
| 3.19 | Создание исполняемого файла . . . . .              | 22 |

## Список таблиц

# 1 Цель работы

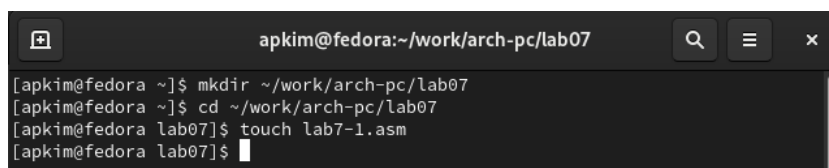
Освоение арифметических инструкций языка ассемблера NASM.

## 2 Задание

Здесь приводится описание задания в соответствии с рекомендациями методического пособия и выданным вариантом.

### 3 Выполнение лабораторной работы

Создаем каталог для программ лабораторной работы №7, переходим в него и создаем файл lab7-1.asm (рис. 3.1)

A terminal window titled 'apkim@fedora:~/work/arch-pc/lab07' with search, menu, and close buttons. The terminal shows the following commands and output:

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

Рис. 3.1: Создание каталога

Введем в файл lab7-1.asm текст программы из листинга 7.1. (рис. 3.2)

```
%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 sprintfLF

    call quit
```

Рис. 3.2: Введем в файл текст программы

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



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

Рис. 3.3: Создание исполняемого файла

Далее изменяем текст программы и вместо символов, запишем в регистры числа. (рис. 3.4)

```
GNU nano 5.8
%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. 3.4: Изменяем текст программы

Создаем исполняемый файл и запускаем его. Этот символ не отображается при выводе на экран, так как символ 10 - символ перевода строки (рис. 3.5)

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

[apkim@fedora lab07]$ █
```

Рис. 3.5: Создание исполняемого файла

Создаем файл lab7-2.asm и вводим в него текст программы из листинга 7.2 (рис. 3.6)

```
GNU nano 5.8
#include 'in_out.asm'
SECTION .text
GLOBAL _start
_start:

mov eax, '6'
mov ebx, '4'
add eax, ebx
call iprintLF

call quit
```

Рис. 3.6: Текст программы из листинга 7.2

Создаем исполняемый файл и запускаем его. (рис. 3.7)

```

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

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

[apkim@fedora lab07]$

```

Рис. 3.7: Создаем исполняемый файл

Аналогично предыдущему примеру изменим символы на числа (рис. 3.8)

```

GNU nano 5.8
%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: Заменяем символы на числа

Создаем исполняемый файл и запускаем его (рис. 3.9)

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

Рис. 3.9: Создаем исполняемый файл и запускаем его

Заменяем функцию `iprintLF` на `iprint` (рис. 3.10)

```
GNU nano 5.8
#include 'in_out.asm'
SECTION .text
GLOBAL _start
_start:

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

Рис. 3.10: Заменяем функцию

Создаем исполняемый файл и запускаем его (рис. 3.11)

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

Рис. 3.11: Создаем исполняемый файл

Создаем файл lab7-3.asm и вводим туда текст программы из листинга 7.3 (рис. 3.12)

```
GNU nano 5.8 /h
%include 'in_out.asm'
SECTION .data

div: DB 'Результат: ',0
rem: DB 'Остаток от деления: ',0

SECTION .text
GLOBAL _start
_start:

; --- Вычисление выражения
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. 3.12: Текст программы

Создаем исполняемый файл и запускаем его (рис. 3.13)

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

Рис. 3.13: Создаем исполняемый файл

Изменяем текст программы для вычисления выражения  $f(x)=(4*6+2)/5$  (рис. 3.14)



```

%include 'in_out.asm'
SECTION .data

div: DB 'Результат: ',0
rem: DB 'Остаток от деления: ',0

SECTION .text
GLOBAL _start
_start:

; --- Вычисление выражения
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

```

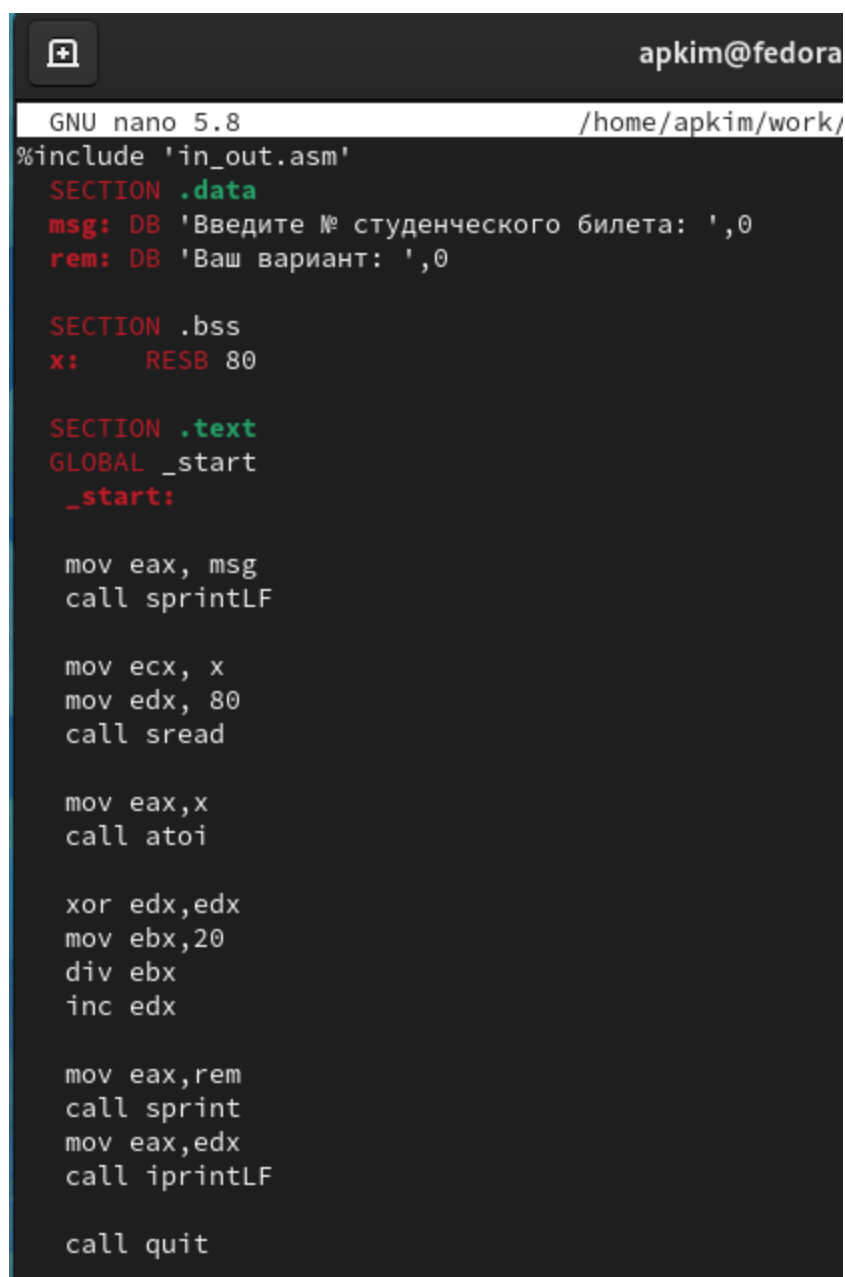
Рис. 3.14: Изменяем текст программы

Создаем исполняемый файл и запускаем его (рис. 3.15)

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

Рис. 3.15: Создаем исполняемый файл

Создаем файл `variant.asm` и вводим в него текст программы из листинга 7.4 (рис. 3.16)



```
GNU nano 5.8 /home/apkim/work/
%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. 3.16: Текст программы из листинга 7.4

Создаем исполняемый файл и запускаем его (рис. 3.17)

```

[apkim@fedora lab07]$ mc
[apkim@fedora lab07]$ nasm -f elf variant.asm
[apkim@fedora lab07]$ ld -m elf_i386 -o variant variant.o
[apkim@fedora lab07]$ ./variant
Введите № студенческого билета:
1132226479
Ваш вариант: 20

```

Рис. 3.17: Создание исполняемого файла

#Ответы на вопросы. 1. Какие строки листинга 7.4 отвечают за вывод на экран сообщения ‘Ваш вариант:’? Ответ: `mov eax,rem; call sprint` 2. Для чего используется следующие инструкции? `nasm mov ecx, x mov edx, 80 call sread` Ответ: Эти инструкции используются для ввода переменной X с клавиатуры и сохранения введенных данных. 3. Для чего используется инструкция “`call atoi`”? Ответ: Эта инструкция используется для преобразование кода переменной ASCII в число. 4. Какие строки листинга 7.4 отвечают за вычисления варианта? Ответ: `xor edx,edx; mov ebx, 20 div ebx inc edx` 5. В какой регистр записывается остаток от деления при выполнении инструкции “`div ebx`”? Ответ: в регистре `edx` 6. Для чего используется инструкция “`inc edx`”? Ответ: Для увеличения значения `edx` на 1. 7. Какие строки листинга 7.4 отвечают за вывод на экран результата вычислений? Ответ: `mov eax,edx call iprintLF`

Задание для самостоятельной работы. У меня выпал вариант 20. Текст программы для функции из варианта 20. (рис. 3.18)

```

GNU nano 5.8
%include 'in_out.asm'

SECTION .data
div: DB 'Результат: ',0
rem: DB 'Остаток от деления: ',0x
f: DB 'x^3*1/3+21',0
x1: DB 'Введите значение x: ',0
otv1: DB 'Ответ при x= ',0

SECTION .bss
x: RESB 80

SECTION .text
GLOBAL _start
_start:

mov eax,f
call sprintfLF

mov eax,x1
call sprintf

mov ecx,x
mov edx,80
call sread

mov eax,x
call atoi
mov ebx,eax
mul ebx
mul ebx
mov ebx,3
div ebx
add eax,21

mov edi,eax

mov eax,otv1
call sprintf
mov eax,edi
call iprintLF

call quit

```

Рис. 3.18: Текст программы вариант-20

Создаем исполняемый файл и запускаем его (рис. 3.19)

```
nasm -f elf 1.asm
[arkim@fedora lab07]$ ld -m elf_i386 -o 1 1.o
[arkim@fedora lab07]$ ./1
x^3*1/3+21
Введите значение x: 3
Ответ при x= 30
[arkim@fedora lab07]$ ./1
x^3*1/3+21
Введите значение x: 1
Ответ при x= 21
```

Рис. 3.19: Создание исполняемого файла

## 4 Выводы

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

## **Список литературы**