

# **Отчёт по лабораторной работе №9**

**Дисциплина: Архитектура компьютера**

**Батова Ирина Сергеевна, НММбд-01-22**

# 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>Задание для самостоятельной работы</b>	<b>15</b>
<b>4</b>	<b>Выводы</b>	<b>18</b>

## Список иллюстраций

2.1	Создание необходимых для работы файлов и каталогов . . . . .	6
2.2	Ввод программы вывода значений регистра есх . . . . .	6
2.3	Запуск программы из файла 'lab9-1.asm' . . . . .	7
2.4	Ввод изменений в файл 'lab9-1.asm' . . . . .	8
2.5	Запуск программы из файла 'lab9-1.asm' с четным числом . . . . .	9
2.6	Повторный ввод изменений в файл 'lab9-1.asm' . . . . .	9
2.7	Запуск измененной программы из файла 'lab9-1.asm' . . . . .	10
2.8	Создание файла 'lab9-2.asm' . . . . .	10
2.9	Ввод программы вывода на экран аргументов командной строки . . . . .	10
2.10	Запуск программы из файла 'lab9-2.asm' . . . . .	11
2.11	Создание файла 'lab9-3.asm' . . . . .	11
2.12	Ввод программы вычисления суммы аргументов командной строки . . . . .	12
2.13	Запуск программы из файла 'lab9-3.asm' . . . . .	12
2.14	Изменение программы в файле 'lab9-3.asm' . . . . .	13
2.15	Запуск измененной программы из файла 'lab9-3.asm' . . . . .	14
3.1	Создание файла 'lab9-4.asm' . . . . .	15
3.2	Ввод программы из задания самостоятельной работы . . . . .	16
3.3	Запуск программы из файла 'lab9-4.asm' . . . . .	17

## **Список таблиц**

# 1 Цель работы

Приобретение навыков написания программ с использованием циклов и обработкой аргументов командной строки.

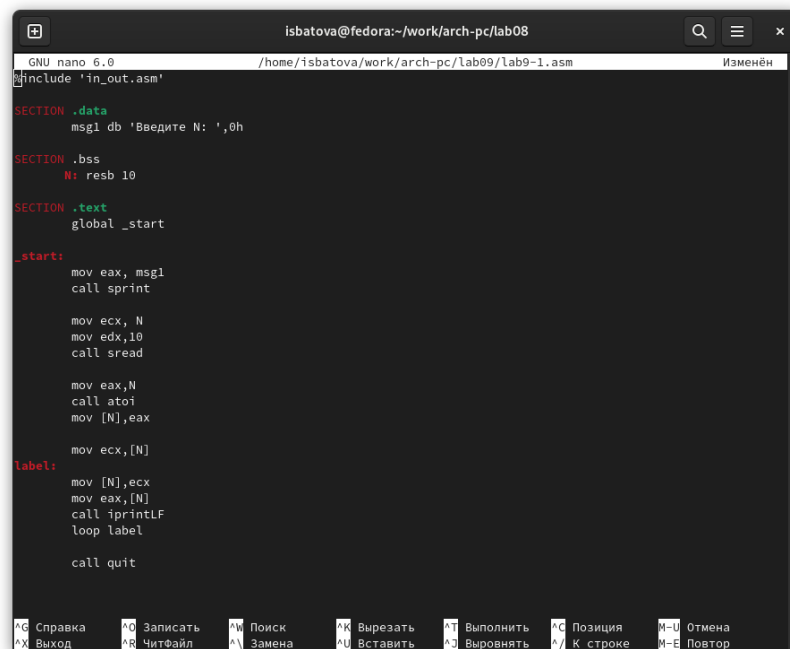
## 2 Выполнение лабораторной работы

1. Для начала работы создаем каталог 'lab09' с помощью команды `mkdir`, переходим в него (команда `cd`) и создаем в нем файл 'lab9-1.asm' (рис. 2.1).

```
[isbatova@fedora ~]$ mkdir ~/work/arch-pc/lab09
[isbatova@fedora ~]$ cd ~/work/arch-pc/lab09
[isbatova@fedora lab09]$ touch lab9-1.asm
```

Рис. 2.1: Создание необходимых для работы файлов и каталогов

Далее открываем файл 'lab9-1.asm' и вводим в него листинг 9.1 из лабораторной работы - программу вывода значений регистра `ecx` (рис. 2.2).



```
GNU nano 6.0 /home/isbatova/work/arch-pc/lab09/lab9-1.asm
#include "in_out.asm"

SECTION .data
    msg1 db 'Введите N: ',0h

SECTION .bss
    N: resb 10

SECTION .text
    global _start

_start:
    mov eax, msg1
    call sprint

    mov ecx, N
    mov edx, 10
    call sread

    mov eax, N
    call atoi
    mov [N], eax

    mov ecx, [N]
label:
    mov [N], ecx
    mov eax, [N]
    call iprintfLF
    loop label

    call quit
```

Рис. 2.2: Ввод программы вывода значений регистра `ecx`

Создаем исполняемый файл и проверяем его работу (рис. 2.3). Программа работает корректно.

```
[isbatova@fedora lab09]$ nasm -f elf lab9-1.asm
[isbatova@fedora lab09]$ ld -m elf_i386 lab9-1.o -o lab9-1
[isbatova@fedora lab09]$ ./lab9-1
Введите N: 3
3
2
1
```

Рис. 2.3: Запуск программы из файла 'lab9-1.asm'

После этого вновь открываем файл 'lab9-1.asm' и немного видоизменяем его - в начале цикла вводим строчку "sub ecx,1" (рис. 2.4).

```

1 %include 'in_out.asm'
2
3 SECTION .data
4     msg1 db 'Введите N: ',0h
5
6 SECTION .bss
7     N: resb 10
8
9 SECTION .text
10    global _start
11
12 _start:
13     mov eax, msg1
14     call sprint
15
16     mov ecx, N
17     mov edx, 10
18     call sread
19
20     mov eax, N
21     call atoi
22     mov [N], eax
23
24     mov ecx, [N]
25 label:
26     sub ecx, 1
27     mov [N], ecx
28     mov eax, [N]
29     call iprintLF
30     loop label
31
32     call quit

```

Рис. 2.4: Ввод изменений в файл 'lab9-1.asm'

Создаем исполняемый файл и проверяем его работу. Если мы вводим четное число (например, 4), программа выводит нечетные числа 3 и 1 (соответственно, число проходов не соответствует введенному числу) (рис. 2.5).



```

[isbatova@fedora lab09]$ nasm -f elf lab9-1.asm
[isbatova@fedora lab09]$ ld -m elf_i386 lab9-1.o -o lab9-1
[isbatova@fedora lab09]$ ./lab9-1
Введите N: 4
3
1
[isbatova@fedora lab09]$ 

```

Рис. 2.5: Запуск программы из файла 'lab9-1.asm' с четным числом

Если выводить нечетное число, у нас получается бесконечный цикл.

Поэтому для корректной работы мы будем использовать стек. Вновь открываем файл 'lab9-1.asm' и вводим в начале цикла 'push ecx', а перед командой 'loop' вводим 'pop ecx' (рис. 2.6).

```

1 %include 'in_out.asm'
2
3 SECTION .data
4     msg1 db 'Введите N: ',0h
5
6 SECTION .bss
7     N: resb 10
8
9 SECTION .text
10    global _start
11
12 _start:
13     mov eax, msg1
14     call sprint
15
16     mov ecx, N
17     mov edx, 10
18     call sread
19
20     mov eax, N
21     call atoi
22     mov [N], eax
23
24     mov ecx, [N]
25 label:
26     push ecx
27     sub ecx, 1
28     mov [N], ecx
29     mov eax, [N]
30     call iprintLF
31     pop ecx
32     loop label
33
34     call quit

```

Рис. 2.6: Повторный ввод изменений в файл 'lab9-1.asm'

Создаем исполняемый файл и проверяем его работу (рис. 2.7). Программа работает корректно и число проходов цикла соответствует введенному числу.

```

[isbatova@fedora lab09]$ nasm -f elf lab9-1.asm
[isbatova@fedora lab09]$ ld -m elf_i386 lab9-1.o -o lab9-1
[isbatova@fedora lab09]$ ./lab9-1
Введите N: 3
2
1
0

```

Рис. 2.7: Запуск измененной программы из файла 'lab9-1.asm'

2. Для дальнейшей работы создаем файл 'lab9-2.asm' (рис. 2.8).

```

[isbatova@fedora lab09]$ touch lab9-2.asm
[isbatova@fedora lab09]$ 

```

Рис. 2.8: Создание файла 'lab9-2.asm'

Открываем файл и вводим в него программу из листинга 9.2 - программу вывода на экран аргументов командной строки (рис. 2.9).

```

1 %include 'in_out.asm'
2
3 SECTION .text
4     global _start
5
6 _start:
7     pop ecx
8     pop edx
9     sub ecx,1
10 next:
11     cmp ecx,0
12     jz _end
13     pop eax
14     call sprintf
15     loop next
16
17 _end:
18     call quit

```

Рис. 2.9: Ввод программы вывода на экран аргументов командной строки

Создаем исполняемый файл и проверяем его работу (рис. 2.10). Программой было обработано три аргумента.

```
[isbatova@fedora lab09]$ nasm -f elf lab9-2.asm
[isbatova@fedora lab09]$ ld -m elf_i386 lab9-2.o -o lab9-2
[isbatova@fedora lab09]$ ./lab9-2 1 2 3
1
2
3
[isbatova@fedora lab09]$
```

Рис. 2.10: Запуск программы из файла 'lab9-2.asm'

3. Далее нам необходимо создать файл 'lab9-3.asm' (рис. 2.11).

```
[isbatova@fedora lab09]$ touch lab9-3.asm
```

Рис. 2.11: Создание файла 'lab9-3.asm'

После этого открываем файл 'lab9-3.asm' и вводим в него программу из листинга 9.2 - программу вычисления суммы аргументов командной строки (рис. 2.12).

```

%include 'in_out.asm'

SECTION .data
    msg db "Результат: ",0

SECTION .text
    global _start

_start:
    pop ecx
    pop edx
    sub ecx,1
    mov esi,0

next:
    cmp ecx,0h
    jz _end
    pop eax
    call atoi
    add esi,eax
    loop next

_end:
    mov eax,msg
    call sprint
    mov eax,esi
    call iprintLF
    call quit

```

Рис. 2.12: Ввод программы вычисления суммы аргументов командной строки

Создаем исполняемый файл и проверяем его работу (рис. 2.13). Проверяем ответ аналитически и понимаем, что программа работает корректно.

```

[isbatova@fedora lab09]$ nasm -f elf lab9-3.asm
[isbatova@fedora lab09]$ ld -m elf_i386 lab9-3.o -o lab9-3
[isbatova@fedora lab09]$ ./lab9-3 12 13 7 10 5
Результат: 47
[isbatova@fedora lab09]$ 

```

Рис. 2.13: Запуск программы из файла 'lab9-3.asm'

Далее нам надо отредактировать файл так, чтобы аргументы перемножались.

Открываем файл 'lab9-3.asm' и вводим изменения (рис. 2.14).

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

```
SECTION .data
```

```
msg db "Результат: ",0
```

```
SECTION .text
```

```
global _start
```

```
_start:
```

```
pop ecx
```

```
pop edx
```

```
sub ecx,1
```

```
mov esi,1
```

```
mov eax,1
```

```
next:
```

```
cmp ecx,0
```

```
jz _end
```

```
pop eax
```

```
call atoi
```

```
mov ebx,eax
```

```
mov eax,esi
```

```
mul ebx
```

```
mov esi,eax
```

```
loop next
```

```
_end:
```

```
mov eax,msg
```

```
call sprint
```

```
mov eax,esi
```

```
call iprintLF
```

```
call quit
```

Рис. 2.14: Изменение программы в файле 'lab9-3.asm'

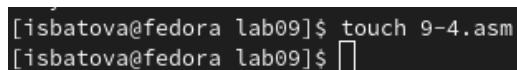
Создаем исполняемый файл и проверяем его работу (рис. 2.15). Проверяем ответ аналитически и понимаем, что программа работает корректно.

```
[isbatova@fedora lab09]$ nasm -f elf lab9-3.asm  
[isbatova@fedora lab09]$ ld -m elf_i386 lab9-3.o -o lab9-3  
[isbatova@fedora lab09]$ ./lab9-3 1 2 3  
Результат: 6
```

Рис. 2.15: Запуск измененной программы из файла 'lab9-3.asm'

### 3 Задание для самостоятельной работы

Для выполнения самостоятельной работы создаем файл 'lab9-4.asm' (рис. 3.1).



```
[isbatova@fedora lab09]$ touch 9-4.asm  
[isbatova@fedora lab09]$
```

Рис. 3.1: Создание файла 'lab9-4.asm'

У меня вариант 11, поэтому программа написана для  $f(x)=15x+2$  (рис. 3.2).

```

%include 'in_out.asm'

SECTION .data
    primer: DB 'Функция: f(x)=15x+2',0
    result: DB 'Результат: ',0

SECTION .text
    global _start

_start:
    mov eax, primer
    call sprintLF

    pop ecx
    pop edx
    sub ecx,1
    mov esi,0

next:
    cmp ecx,0h
    jz _end
    pop eax
    call atoi
    mov ebx,15
    mul ebx
    add eax,2
    add esi,eax
    loop next

_end:
    mov eax,result
    call sprint
    mov eax,esi
    call iprintLF
    call quit

```

Рис. 3.2: Ввод программы из задания самостоятельной работы

Сохраняем файл, создаем исполняемый файл и проверяем его работу (рис. 3.3).



Проверяем ответ аналитически и понимаем, что программа работает корректно.

```
[isbatova@fedora lab09]$ nasm -f elf lab9-4.asm
[isbatova@fedora lab09]$ ld -m elf_i386 lab9-4.o -o lab9-4
[isbatova@fedora lab09]$ ./lab9-4 1 2 3 4
Функция:  $f(x)=15x+2$ 
Результат: 158
```

Рис. 3.3: Запуск программы из файла 'lab9-4.asm'

## **4 Выводы**

В данной лабораторной работе мной были приобретены навыки написания программ с использованием циклов и обработкой аргументов командной строки.