

Отчет по лабораторной работе №9.

Дисциплина: Архитектура компьютеров.

Лобанова Полина Иннокентьевна

Содержание

1	Цель работы	3
2	Выполнение лабораторной работы	4
3	Выполнение самостоятельной работы	10
4	Выводы	12

1 Цель работы

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

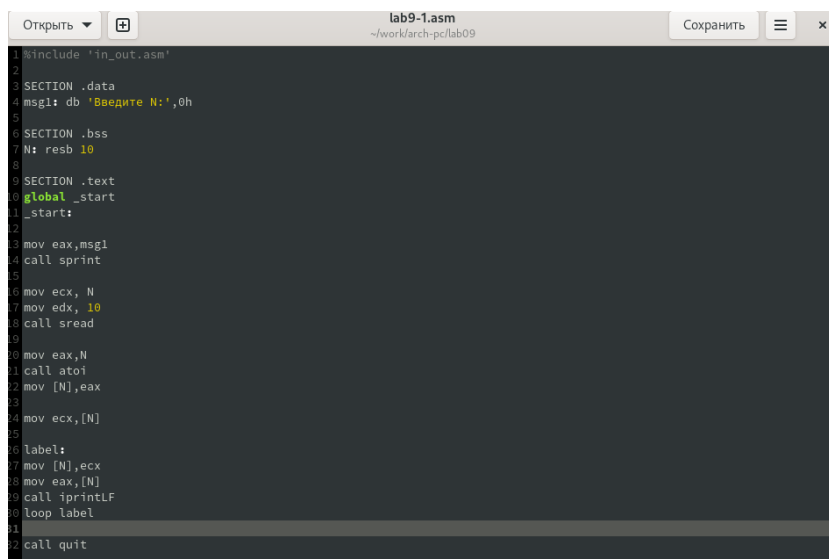
2 Выполнение лабораторной работы

1. Создадим каталог lab09 и текстовый файл lab9-1.asm.

```
[pilobanova@i0 ~]$ mkdir ~/work/arch-pc/lab09
[pilobanova@i0 ~]$ cd ~/work/arch-pc/lab09
[pilobanova@i0 lab09]$ touch lab9-1.asm
[pilobanova@i0 lab09]$
```

Рис. 2.1: Создание каталога lab09 и файла lab9-1.asm.

2. Заполним файл.



```
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 _start:
12
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
26 label:
27 mov [N],ecx
28 mov eax,[N]
29 call iprintf
30 loop label
31
32 call quit
```

Рис. 2.2: Текст программы в файле lab9-1.asm.

3. Создадим исполняемый файл и проверим его работу.

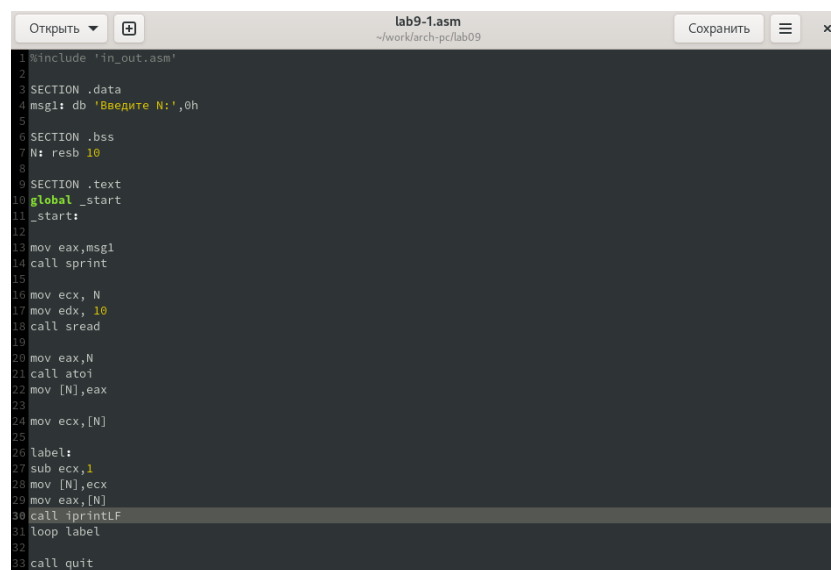
```

[pilobanova@10 lab09]$ nasm -f elf lab9-1.asm
[pilobanova@10 lab09]$ ld -m elf_i386 -o lab9-1 lab9-1.o
[pilobanova@10 lab09]$ ./lab9-1
Введите N:5
5
4
3
2
1
[pilobanova@10 lab09]$ ./lab9-1
Введите N:9
9
8
7
6
5
4
3
2
1
[pilobanova@10 lab09]$

```

Рис. 2.3: Создание исполняемого файла и его запуск.

4. Изменим текст программы, добавив изменение значение регистра ecx в цикле.



```

lab9-1.asm
~/work/arch-pc/lab09
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 _start:
12
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
26 label:
27 sub ecx,1
28 mov [N],ecx
29 mov eax,[N]
30 call iprintLF
31 loop label
32
33 call quit

```

Рис. 2.4: Измененный текст программы в файле lab9-1.asm.

5. Создадим исполняемый файл и проверим его работу. В результате чего цикл стал бесконечным.

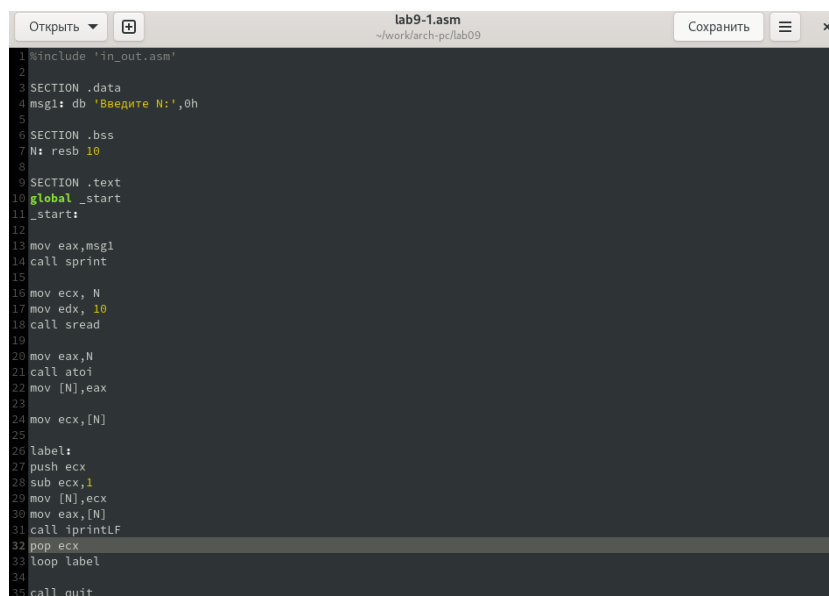
```
[pilobanova@i0 lab09]$ nasm -f elf lab9-1.asm
[pilobanova@i0 lab09]$ ld -m elf_i386 -o lab9-1 lab9-1.o
[pilobanova@i0 lab09]$ ./lab9-1
Введите N:5
```

Рис. 2.5: Создание исполняемого файла и его запуск.

```
4294960582
4294960580
4294960578
4294960576
4294960574
4294960572
4294960570
4294960568
4294960566
4294960564
4294960562
4294960560
4294960558
4294960556
4294960554
4294960552
4294960550
4294960548
4294960546
4294960544
4294960542
4294960540
4294960538
```

Рис. 2.6: Результат запуска.

6. Изменим текст программы, добавив команды `push` и `pop`.



```
lab9-1.asm
~/work/arch-pc/lab09

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 _start:
12
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
26 label:
27 push ecx
28 sub ecx,1
29 mov [N],ecx
30 mov eax,[N]
31 call iprintLF
32 pop ecx
33 loop label
34
35 call quit
```

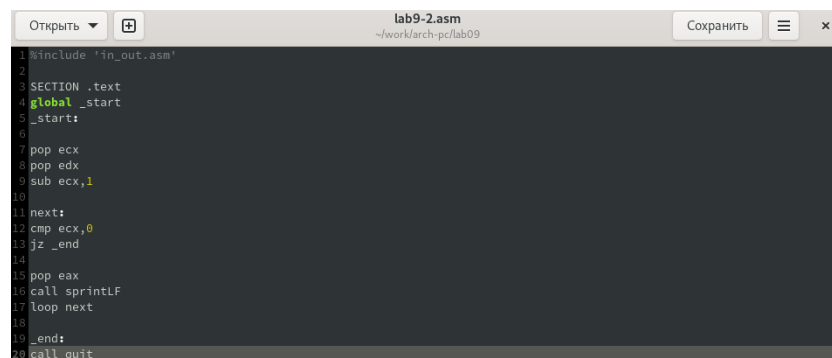
Рис. 2.7: Измененный текст программы в файле `lab9-1.asm`.

7. Создадим исполняемый файл и проверим его работу. В результате мы получили, что число проходов цикла соответствует значению, введенному с клавиатуры.

```
[pilobanova@10 lab09]$ nasm -f elf lab9-1.asm
[pilobanova@10 lab09]$ ld -m elf_i386 -o lab9-1 lab9-1.o
[pilobanova@10 lab09]$ ./lab9-1
Введите N:5
4
3
2
1
0
[pilobanova@10 lab09]$ ./lab9-1
Введите N:9
8
7
6
5
4
3
2
1
0
[pilobanova@10 lab09]$
```

Рис. 2.8: Создание исполняемого файла и его запуск.

8. Создадим текстовый файл lab9-2.asm и заполним его.



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

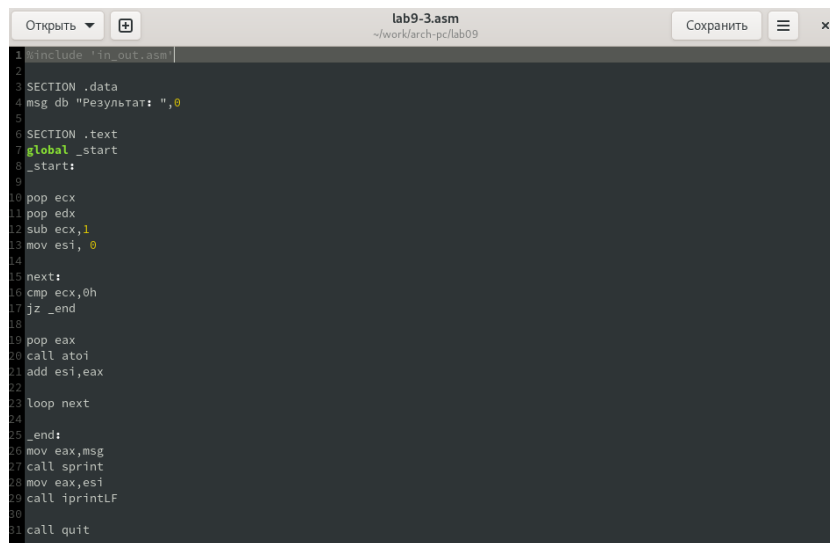
Рис. 2.9: Текст программы в файле lab9-2.asm.

9. Создадим исполняемый файл и проверим его работу.

```
[pilobanova@10 lab09]$ nasm -f elf lab9-2.asm
[pilobanova@10 lab09]$ ld -m elf_i386 -o lab9-2 lab9-2.o
[pilobanova@10 lab09]$ ./lab9-2 аргумент1 аргумент 2 'аргумент 3'
аргумент1
аргумент
2
аргумент 3
```

Рис. 2.10: Создание исполняемого файла и его запуск.

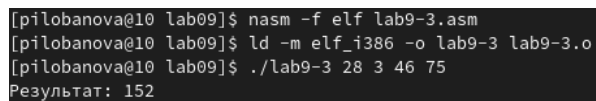
10. Создадим текстовый файл lab9-3.asm и заполним его.



```
1 %include "in_out.asm"
2
3 SECTION .data
4 msg db "Результат: ",0
5
6 SECTION .text
7 global _start
8 _start:
9
10 pop ecx
11 pop edx
12 sub ecx,1
13 mov esi, 0
14
15 next:
16 cmp ecx,0h
17 jz _end
18
19 pop eax
20 call atoi
21 add esi,eax
22
23 loop next
24
25 _end:
26 mov eax,msg
27 call sprint
28 mov eax,esi
29 call iprintf
30
31 call quit
```

Рис. 2.11: Текст программы в файле lab9-3.asm.

11. Создадим исполняемый файл и проверим его работу.



```
[pilobanova@i0 lab09]$ nasm -f elf lab9-3.asm
[pilobanova@i0 lab09]$ ld -m elf_i386 -o lab9-3 lab9-3.o
[pilobanova@i0 lab09]$ ./lab9-3 28 3 46 75
Результат: 152
```

Рис. 2.12: Создание исполняемого файла и его запуск.

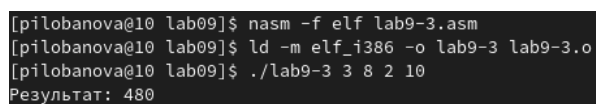
12. Изменим текст программы для вычисления произведения аргументов командной строки.



```
1 %include "in_out.asm"
2
3 SECTION .data
4 msg db "Результат: ",0
5
6 SECTION .text
7 global _start
8 _start:
9
10 pop ecx
11 pop edx
12 sub ecx,1
13 mov esi,1
14
15 next:
16 cmp ecx,0h
17 jz _end
18
19 pop eax
20 call atoi
21 mul esi
22 mov esi,eax
23
24 loop next
25
26 _end:
27 mov eax,msg
28 call sprint
29 mov eax,esi
30 call iprintf
31
32 call quit
```

Рис. 2.13: Измененный текст программы в файле *lab9-3.asm*.

13. Создадим исполняемый файл и проверим его работу.



```
[pilobanova@i0 lab09]$ nasm -f elf lab9-3.asm
[pilobanova@i0 lab09]$ ld -m elf_i386 -o lab9-3 lab9-3.o
[pilobanova@i0 lab09]$ ./lab9-3 3 8 2 10
Результат: 480
```

Рис. 2.14: Создание исполняемого файла и его запуск.

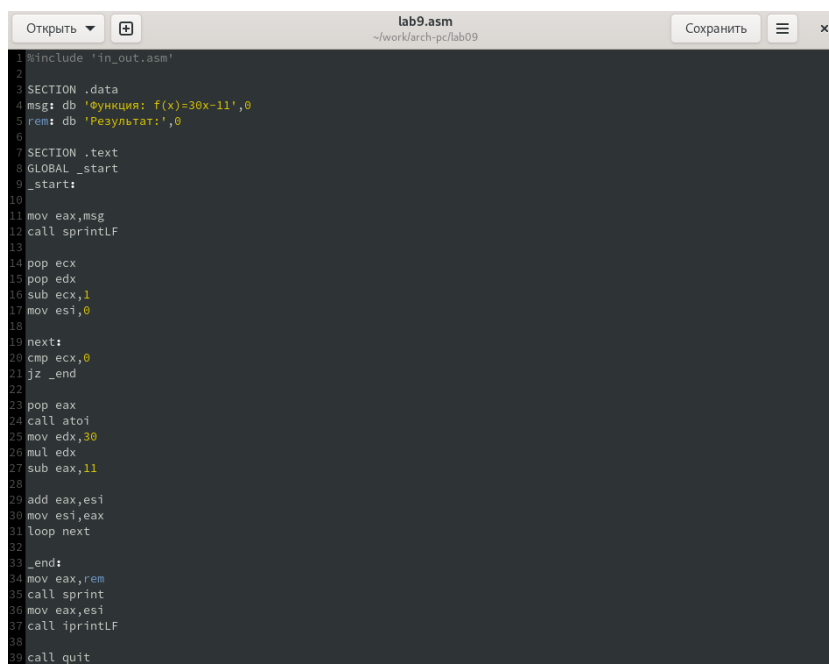
3 Выполнение самостоятельной работы

1. Создадим текстовый файл lab9.asm.

```
[pilobanova@i10 lab09]$ touch lab9.asm
```

Рис. 3.1: Создание файла lab9.asm.

2. Заполним файл так, чтобы программа находила сумму значений функции (выбранной в соответствии с вариантом, полученным при выполнении лабораторной работы №7) $f(x)=30x-11$ для $x=x_1, x_2, x_3, \dots, x_n$.



```
1 %include 'in_out.asm'
2
3 SECTION .data
4 msg: db 'Функция: f(x)=30x-11',0
5 rem: db 'Результат:',0
6
7 SECTION .text
8 GLOBAL _start
9 _start:
10
11 mov eax,msg
12 call sprintf
13
14 pop ecx
15 pop edx
16 sub ecx,1
17 mov esi,0
18
19 next:
20 cmp ecx,0
21 jz _end
22
23 pop eax
24 call atoi
25 mov edx,30
26 mul edx
27 sub eax,11
28
29 add eax,esi
30 mov esi,eax
31 loop next
32
33 _end:
34 mov eax,rem
35 call sprintf
36 mov eax,esi
37 call iprintf
38
39 call quit
```

Рис. 3.2: Текст программы в файле lab9.asm.

3. Создадим исполняемый файл и проверим его работу.

```
[pilobanova@10 lab09]$ nasm -f elf lab9.asm
[pilobanova@10 lab09]$ ld -m elf_i386 -o lab9 lab9.o
[pilobanova@10 lab09]$ ./lab9 1 2 3 4
Функция:  $f(x)=30x-11$ 
Результат:256
```

Рис. 3.3: Создание исполняемого файла и его запуск.

4 Выводы

Я научилась писать программы с использованием циклов и обработкой аргументов командной строки.