

Отчёта по лабораторной работе №8

Программирование цикла. Обработка аргументов командной строки.

Сидорова Арина Валерьевна

Содержание

1	Цель работы	4
2	Выполнение лабораторной работы	5
2.1	Реализация циклов в NASM	5
2.2	Обработка аргументов командной строки.	9
2.3	Задание для самостоятельной работы	13
3	Выводы	15

Список иллюстраций

2.1	Создаем каталог с помощью команды <code>mkdir</code> и файл с помощью команды <code>touch</code>	5
2.2	Заполняем файл	6
2.3	Запускаем файл и проверяем его работу	6
2.4	Изменяем файл	7
2.5	Запускаем файл и смотрим на его работу	7
2.6	Редактируем файл	8
2.7	Проверяем, сошелся ли наш вывод с данным в условии выводом .	9
2.8	Создаем файл командой <code>touch</code>	9
2.9	Заполняем файл	10
2.10	Смотрим на работу программ	10
2.11	Создаем файл командой <code>touch</code>	11
2.12	Заполняем файл	11
2.13	Смотрим на работу программы	11
2.14	Изменяем файл	12
2.15	Проверяем работу файла(работает правильно)	12
2.16	Создаем файл командой <code>touch</code>	13
2.17	Пишем программу	13
2.18	Смотрим на работу программы	14
2.19	Смотрим на работу программы	14

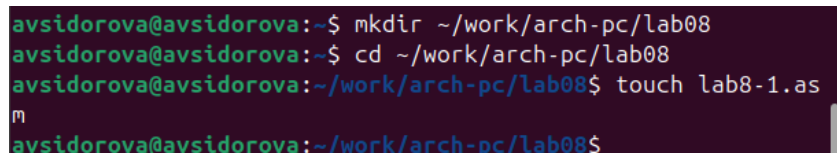
1 Цель работы

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

2 Выполнение лабораторной работы

2.1 Реализация циклов в NASM

Создаем каталог для программ ЛБ8, и в нем создаем файл (рис. fig. 2.1).



```
avsidorova@avsidorova:~$ mkdir ~/work/arch-pc/lab08
avsidorova@avsidorova:~$ cd ~/work/arch-pc/lab08
avsidorova@avsidorova:~/work/arch-pc/lab08$ touch lab8-1.as
m
avsidorova@avsidorova:~/work/arch-pc/lab08$
```

Рис. 2.1: Создаем каталог с помощью команды `mkdir` и файл с помощью команды `touch`

Открываем файл в Midnight Commander и заполняем его в соответствии с листингом 8.1 (рис. fig. 2.2).

```
/home/avsidorova/work/arch-pc/lab08/lab8-1.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 iprintLF
    loop label
    call quit
```

^G Справка ^O Записать ^W Поиск ^K Вырезать
^X Выход ^R ЧитФайл ^\ Замена ^U Вставить

Рис. 2.2: Заполняем файл

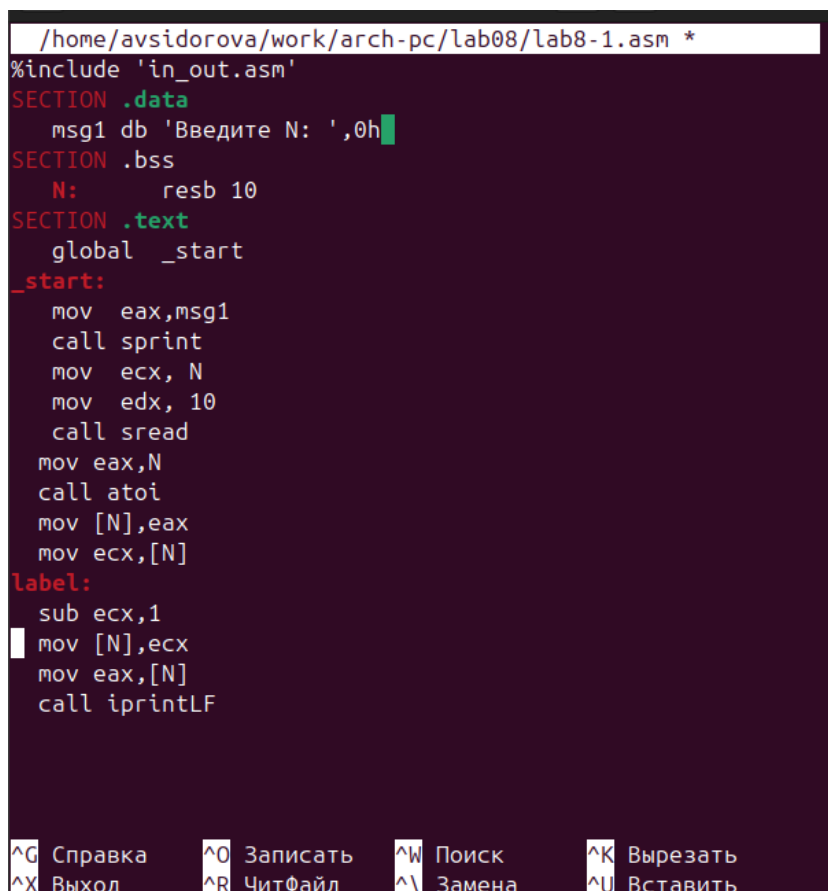
Создаем исполняемый файл и запускаем его (рис. fig. 2.3).

```
avsidorova@avsidorova:~/work/arch-pc/lab08$ nasm -f elf lab8-1.asm
avsidorova@avsidorova:~/work/arch-pc/lab08$ ld -m elf_i386 -o lab8-1 lab8-1.o
avsidorova@avsidorova:~/work/arch-pc/lab08$ ./lab8-1
Введите N: 8
8
7
6
5
4
3
2
1
avsidorova@avsidorova:~/work/arch-pc/lab08$
```

Рис. 2.3: Запускаем файл и проверяем его работу

Снова открываем файл для редактирования и изменяем его, добавив измене-

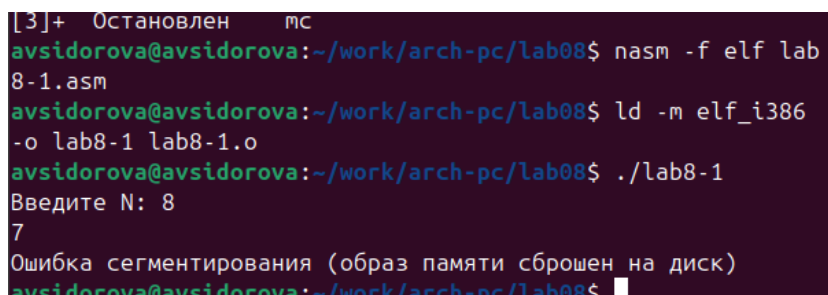
ние значения регистра в цикле (рис. fig. 2.4).



```
/home/avsidorova/work/arch-pc/lab08/lab8-1.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:
    sub ecx,1
    mov [N],ecx
    mov eax,[N]
    call iprintLF
```

Рис. 2.4: Изменяем файл

Создаем исполняемый файл и запускаем его (рис. fig. 2.5).



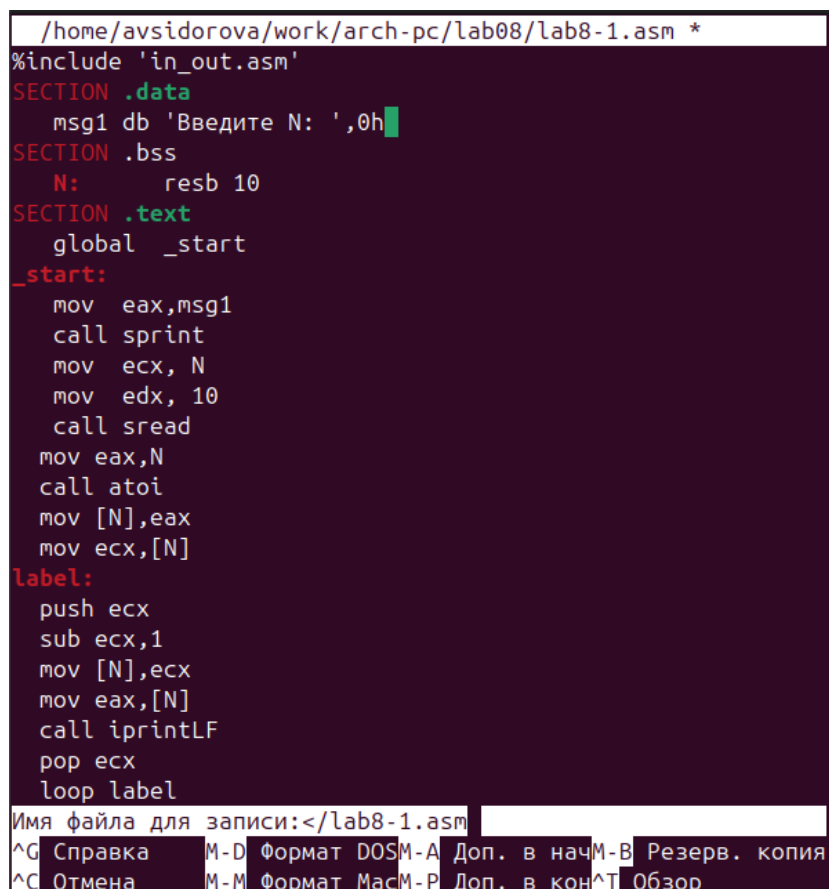
```
[3]+ Остановлен  ms
avsidorova@avsidorova:~/work/arch-pc/lab08$ nasm -f elf lab
8-1.asm
avsidorova@avsidorova:~/work/arch-pc/lab08$ ld -m elf_i386
-o lab8-1 lab8-1.o
avsidorova@avsidorova:~/work/arch-pc/lab08$ ./lab8-1
Введите N: 8
7
Ошибка сегментирования (образ памяти сброшен на диск)
avsidorova@avsidorova:~/work/arch-pc/lab08$
```

Рис. 2.5: Запускаем файл и смотрим на его работу

Регистр ecx принимает значения 9,7,5,3,1(на вход подается число 10, в цикле label данный регистр уменьшается на 2 командой sub и loop).

Число проходов цикла не соответствует числу N, так как уменьшается на 2.

Снова открываем файл для редактирования и изменяем его, чтобы все корректно работало (рис. fig. 2.6).



```
/home/avsidorova/work/arch-pc/lab08/lab8-1.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:
    push ecx
    sub ecx,1
    mov [N],ecx
    mov eax,[N]
    call iprintLF
    pop ecx
    loop label
Имя файла для записи: </lab8-1.asm
^G Справка M-D Формат DOSM-A Доп. в начM-B Резерв. копия
^C Отмена M-M Формат MacM-P Доп. в кон^T Обзор
```

Рис. 2.6: Редактируем файл

Создаем исполняемый файл и запускаем его (рис. fig. 2.7).


```

[2]+  Остановлен    мс
avsidorova@avsidorova:~/work/arch-pc/lab08$ nasm -f elf lab8-1.asm
avsidorova@avsidorova:~/work/arch-pc/lab08$ ld -m elf_i386 -o lab8-1 lab8-1.o
avsidorova@avsidorova:~/work/arch-pc/lab08$ ./lab8-1
Введите N: 8
7
6
5
4
3
2
1
0
Ошибка сегментирования (образ памяти сброшен на диск)
avsidorova@avsidorova:~/work/arch-pc/lab08$

```

Рис. 2.7: Проверяем, сошелся ли наш вывод с данным в условии выводом

В данном случае число проходов цикла равна числу N.

2.2 Обработка аргументов командной строки.

Создаем новый файл (рис. fig. 2.8).

```

avsidorova@avsidorova:~/work/arch-pc/lab08$ touch lab8-2.asm
avsidorova@avsidorova:~/work/arch-pc/lab08$

```

Рис. 2.8: Создаем файл командой touch

Открываем файл в Midnight Commander и заполняем его в соответствии с листингом 8.2 (рис. fig. 2.9).

```
/home/avsidorova/work/arch-pc/lab08/lab8-2.asm *
%include 'in_out.asm'
SECTION .text
global _start
_start:
    pop ecx
    pop edx
    sub ecx,1
next:
    cmp ecx,0
    jz _end
    pop eax
    call sprintLF
    loop next
_end:
    call quit
```

^G Справка ^O Записать ^W Поиск ^K Вырезать
^X Выход ^R ЧитФайл ^\ Замена ^U Вставить

Рис. 2.9: Заполняем файл

Создаем исполняемый файл и проверяем его работу, указав аргументы (рис. fig. 2.10).

```
avsidorova@avsidorova:~/work/arch-pc/lab08$ nasm -f elf lab8-2.asm
avsidorova@avsidorova:~/work/arch-pc/lab08$ ld -m elf_i386 -o lab8-2 lab8-2.o
avsidorova@avsidorova:~/work/arch-pc/lab08$ ./lab8-2 3 8 '10'
```

Рис. 2.10: Смотрим на работу программ

Программой было обработано 3 аргумента.

Создаем новый файл lab8-3.asm (рис. fig. 2.11).

```

10
avsidorova@avsidorova:~/work/arch-pc/lab08$ touch lab8-3.asm
avsidorova@avsidorova:~/work/arch-pc/lab08$

```

Рис. 2.11: Создаем файл командой touch

Открываем файл и заполняем его в соответствии с листингом 8.3 (рис. fig. 2.12).

```

/home/avsidorova-8/lab8-3.asm 318/318 100%
#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: Заполняем файл

Создаём исполняемый файл и запускаем его, указав аргументы (рис. fig. 2.13).

```

avsidorova@avsidorova:~/work/arch-pc/lab08$ nasm -f elf lab8-3.asm
avsidorova@avsidorova:~/work/arch-pc/lab08$ ld -m elf_i386 -o lab8-3 lab8-3.o
avsidorova@avsidorova:~/work/arch-pc/lab08$ ./lab8-3 12 13 7 10 5
Результат: 47
avsidorova@avsidorova:~/work/arch-pc/lab08$

```

Рис. 2.13: Смотрим на работу программы

Снова открываем файл для редактирования и изменяем его, чтобы вычислялось произведение вводимых значений (рис. fig. 2.14).

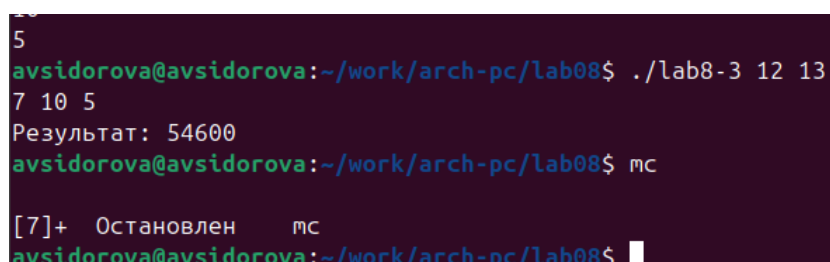


```
/home/avsidorova/work/arch-pc/lab08/lab8-3.asm *
#include 'in_out.asm'
SECTION .data
msg db "Результат: ",0
SECTION .text
global _start
_start:
    pop ecx
    pop edx
    sub ecx,1
    mov esi, 1
next:
    cmp ecx,0h
    jz _end
    pop eax
    call atoi
    mul esi
    mov esi,eax
    loop next
_end:
    mov eax,msg
    call sprint
    mov eax,esi
    call iprintLF
    call quit
```

^G Справка ^O Записать ^W Поиск ^K Вырезать
^X Выход ^R ЧитФайл ^\ Замена ^U Вставить

Рис. 2.14: Изменяем файл

Создаём исполняемый файл и запускаем его, указав аргументы (рис. fig. 2.15).



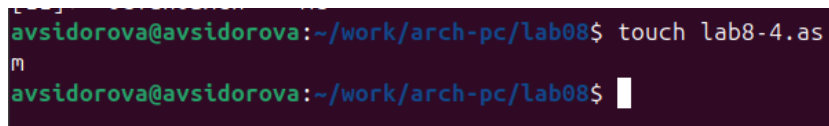
```
5
avsidorova@avsidorova:~/work/arch-pc/lab08$ ./lab8-3 12 13
7 10 5
Результат: 54600
avsidorova@avsidorova:~/work/arch-pc/lab08$ mc
[7]+ Остановлен    mc
avsidorova@avsidorova:~/work/arch-pc/lab08$
```

Рис. 2.15: Проверяем работу файла(работает правильно)

2.3 Задание для самостоятельной работы

ВАРИАНТ-13

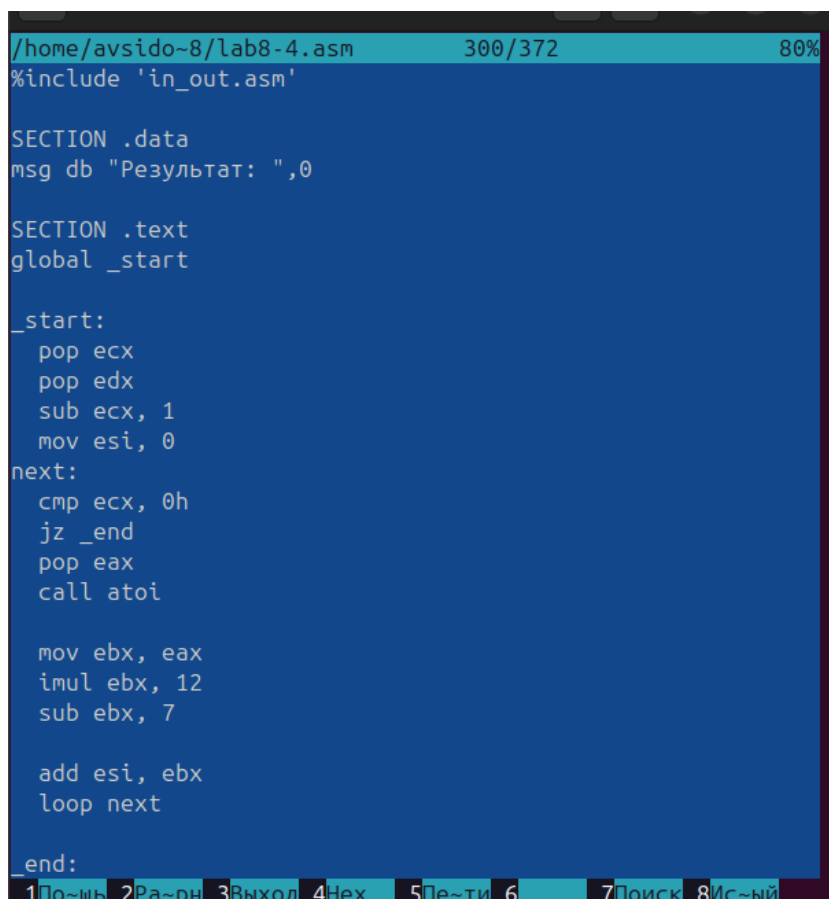
1. Создаем новый файл (рис. fig. 2.16).



```
avsidorova@avsidorova:~/work/arch-pc/lab08$ touch lab8-4.asm
avsidorova@avsidorova:~/work/arch-pc/lab08$
```

Рис. 2.16: Создаем файл командой touch

Открываем его и пишем программу, которая выведет сумму значений, получившихся после решения выражения $12x-7$ (рис. fig. 2.17).



```
/home/avsidorova/lab8-4.asm 300/372 80%
%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

    mov ebx, eax
    imul ebx, 12
    sub ebx, 7

    add esi, ebx
    loop next

_end:
```

Рис. 2.17: Пишем программу

Транслируем файл и смотрим на работу программы (рис. fig. 2.18).

```

[15]+  Остановлен      mc
avsidorova@avsidorova:~/work/arch-pc/lab08$ nasm -f elf lab
8-4.asm
avsidorova@avsidorova:~/work/arch-pc/lab08$ ld -m elf_i386
-o lab8-4 lab8-4.o
avsidorova@avsidorova:~/work/arch-pc/lab08$ ./lab8-4 1 2 3
4
Результат: 92

```

Рис. 2.18: Смотрим на рабботу программы

Транслируем файл и смотрим на работу программы (рис. fig. 2.19).

```

Результат: 92
avsidorova@avsidorova:~/work/arch-pc/lab08$ mc\
>

[16]+  Остановлен      mc
avsidorova@avsidorova:~/work/arch-pc/lab08$ ./lab8-4 4 6 8
Результат: 195
avsidorova@avsidorova:~/work/arch-pc/lab08$

```

Рис. 2.19: Смотрим на рабботу программы

3 Выводы

Приобрели навыки написания программ с использованием циклов и обработкой аргументов командной строки.