

Отчёта по лабораторной работе №8

Программирование цикла. Обработка аргументов командной строки.

Хохлачева Полина Дмитриевна

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Выводы	13

Список иллюстраций

2.1	В каталоге создаём файл	6
2.2	Заполняем файл	6
2.3	Запускаем файл и проверяем работу	7
2.4	Изменяем файл	7
2.5	Смотрим на работу файла	8
2.6	Изменение файла	8
2.7	Проверяем вывод	8
2.8	Создаём командой touch	9
2.9	Заполняем файл	9
2.10	Смотрим работу программы	9
2.11	Новый файл	10
2.12	Заполняем файл	10
2.13	Смотрим на работу программы	10
2.14	Изменяем файл	11
2.15	Проверяем работу	11
2.16	Создаём файл командой touch	12
2.17	Создаём файл командой touch	12

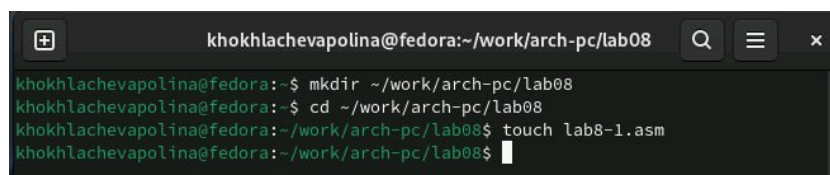
Список таблиц

1 Цель работы

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

2 Выполнение лабораторной работы

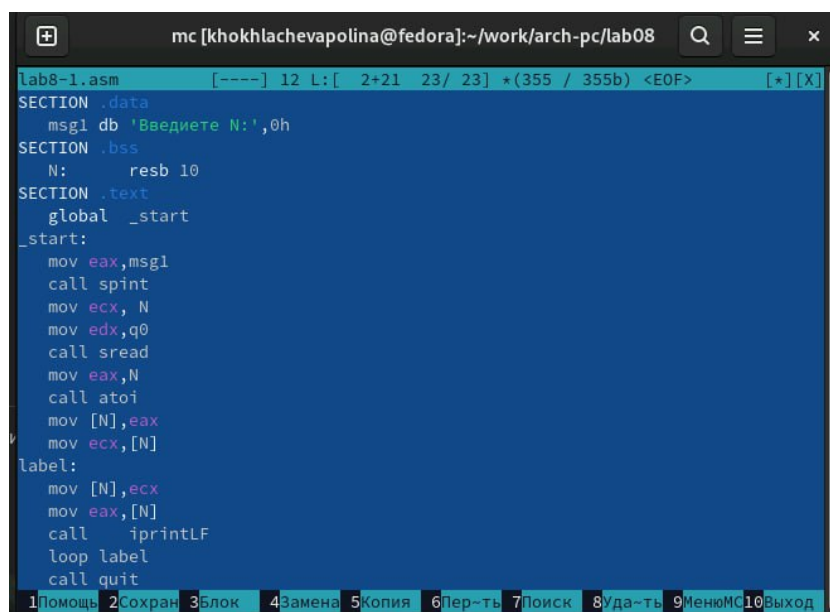
Создаём каталог для лабораторной работы



```
khokhlachevapolina@fedora:~/work/arch-pc/lab08
khokhlachevapolina@fedora:~$ mkdir ~/work/arch-pc/lab08
khokhlachevapolina@fedora:~$ cd ~/work/arch-pc/lab08
khokhlachevapolina@fedora:~/work/arch-pc/lab08$ touch lab8-1.asm
khokhlachevapolina@fedora:~/work/arch-pc/lab08$
```

Рис. 2.1: В каталоге создаём файл

Открываем файл и заполняем его в соответствии с листингом



```
lab8-1.asm  [----] 12 L: [ 2+21 23/ 23] +(355 / 355b) <EOF> [*][X]
SECTION .data
    msg1 db 'Введите N:',0h
SECTION .bss
    N:    resb 10
SECTION .text
    global _start
_start:
    mov eax,msg1
    call spint
    mov ecx, N
    mov edx,q0
    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
```

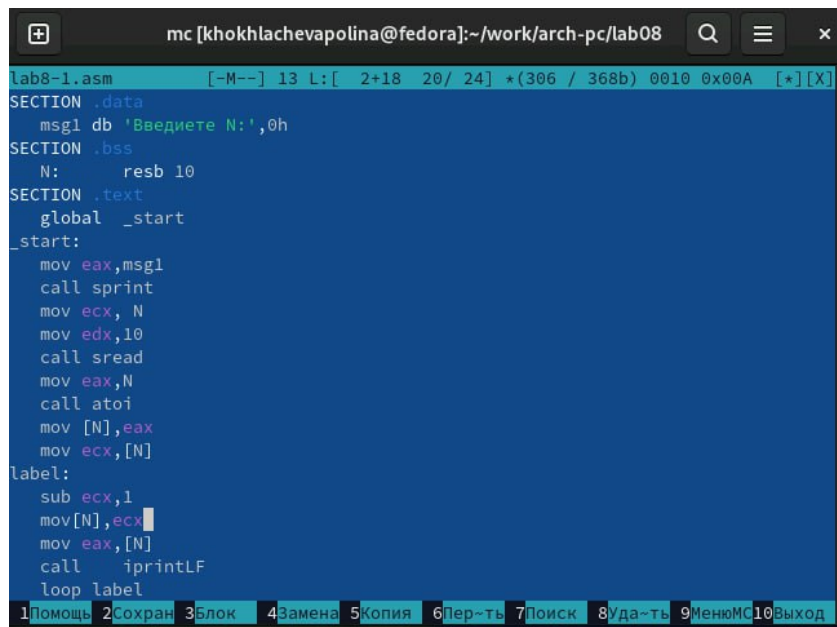
Рис. 2.2: Заполняем файл

Создаём файл и запускаем его

```
khokhlachevapolina@fedora:~/work/arch-pc/lab08$ nasm -f elf lab8-1.asm
khokhlachevapolina@fedora:~/work/arch-pc/lab08$ ld -m elf_i386 -o lab8-1 lab8-1.o
khokhlachevapolina@fedora:~/work/arch-pc/lab08$ ./lab8-1
Введите N:10
10
9
8
7
6
5
4
3
2
1
khokhlachevapolina@fedora:~/work/arch-pc/lab08$
```

Рис. 2.3: Запускаем файл и проверяем работу

Открываем файл и редактируем его



```
mc [khokhlachevapolina@fedora]:~/work/arch-pc/lab08
lab8-1.asm [-M--] 13 L: [ 2+18 20/ 24] *(306 / 368b) 0010 0x00A [*][X]
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
    loop label
1Помощь 2Сохран 3Блок 4Замена 5Копия 6Пер-ть 7Поиск 8Уда-ть 9МенюMC10Выход
```

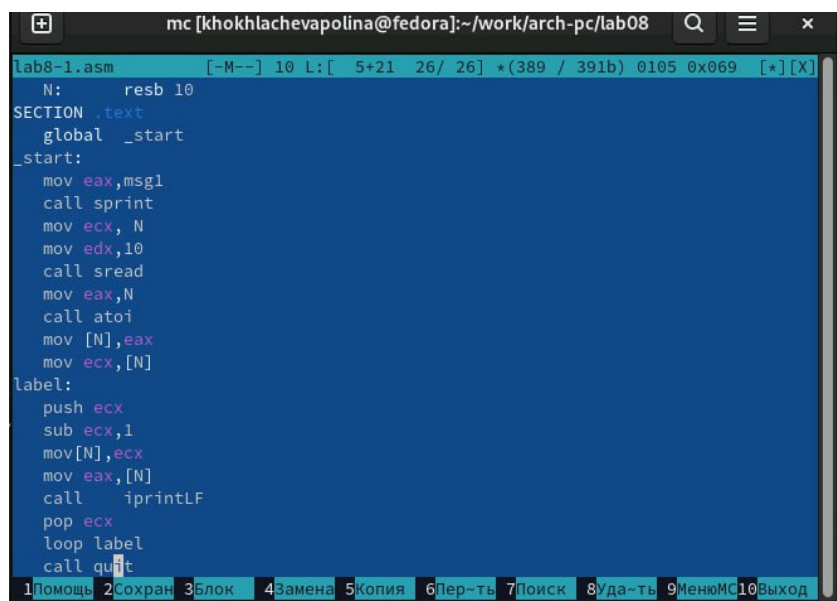
Рис. 2.4: Изменяем файл

Создаём файл и запускаем его

```
khokhlachevapolina@fedora:~/work/arch-pc/lab08$ nasm -f elf lab8-1.asm
khokhlachevapolina@fedora:~/work/arch-pc/lab08$ ld -m elf_i386 -o lab8-1 lab8-1.o
khokhlachevapolina@fedora:~/work/arch-pc/lab08$ ./lab8-1
Введите N:10
9
7
5
3
1
khokhlachevapolina@fedora:~/work/arch-pc/lab08$
```

Рис. 2.5: Смотрим на работу файла

Редактируем файл



```
lab8-1.asm  [-M--] 10 L: [ 5+21 26/ 26] *(389 / 391b) 0105 0x069 [*][X]
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
    call quit
```

Рис. 2.6: Изменение файла

Создаём файл и запускаем его

```
khokhlachevapolina@fedora:~/work/arch-pc/lab08$ nasm -f elf lab8-1.asm
khokhlachevapolina@fedora:~/work/arch-pc/lab08$ ld -m elf_i386 -o lab8-1 lab8-1.o
khokhlachevapolina@fedora:~/work/arch-pc/lab08$ ./lab8-1
Введите N:10
9
8
7
6
5
4
3
2
1
0
khokhlachevapolina@fedora:~/work/arch-pc/lab08$
```

Рис. 2.7: Проверяем вывод

Создаём новый файл

```
4
1
0
khokhlachevapolina@fedora:~/work/arch-pc/lab08$ touch lab8-2.asm
khokhlachevapolina@fedora:~/work/arch-pc/lab08$
```

Рис. 2.8: Создаём командой touch

Открываем файл и заполняем его в соответствии с листингом

```
mc [khokhlachevapolina@fedora:~/work/arch-pc/lab08]
lab8-2.asm [----] 12 L: [ 1+14 15/ 15] *(183 / 183b) <EOF> [*][X]
#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 sprintf
    loop next
_end:
    call quit
```

Рис. 2.9: Заполняем файл

Создаём файл и проверяем его работу

```
khokhlachevapolina@fedora:~/work/arch-pc/lab08$ nasm -f elf lab8-2.asm
khokhlachevapolina@fedora:~/work/arch-pc/lab08$ ld -m elf_i386 -o lab8-2 lab8-2.o
khokhlachevapolina@fedora:~/work/arch-pc/lab08$ ./lab8-2 1 2 '3'
1
23
khokhlachevapolina@fedora:~/work/arch-pc/lab08$ ./lab8-2 1 2 '3'
1
2
3
khokhlachevapolina@fedora:~/work/arch-pc/lab08$
```

Рис. 2.10: Смотрим работу программы

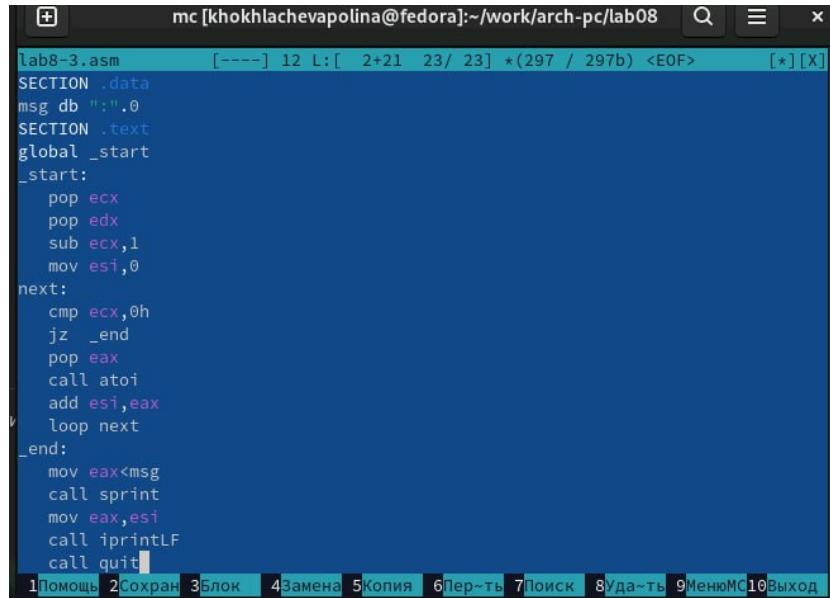
Программой было обработано 3 аргумента

Создаём новый файл

```
khokhlachevapolina@fedora:~/work/arch-pc/lab08$ touch lab8-3.asm
khokhlachevapolina@fedora:~/work/arch-pc/lab08$
```

Рис. 2.11: Новый файл

Открываем файл и заполняем в соответствии с листингом



```
mc [khokhlachevapolina@fedora]:~/work/arch-pc/lab08
lab8-3.asm  [----] 12 L:[ 2+21 23/ 23] *(297 / 297b) <EOF>  [*] [X]
SECTION .data
msg db " ".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 файл и запускаем его

```
khokhlachevapolina@fedora:~/work/arch-pc/lab08$ nasm -f elf lab8-3.asm
khokhlachevapolina@fedora:~/work/arch-pc/lab08$ ld -m elf_i386 -o lab8-3 lab8-3.o
khokhlachevapolina@fedora:~/work/arch-pc/lab08$ ./lab8-3 5 3 4
Результат:12
khokhlachevapolina@fedora:~/work/arch-pc/lab08$
```

Рис. 2.13: Смотрим на работу программы

Открываем файл и редактируем его

```

lab8-3.asm  [----] 14 L: [ 2+15 17/ 24] *(232 / 326b) 0010 0x00A [*] [X]
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
    mul esi
    mov esi,eax
    loop next
_end:
    mov eax,msg
    call sprint
    mov eax,esi
    call iprintLF
1Помощь 2Сохран 3Блок 4Замена 5Копия 6Пер-ть 7Поиск 8Уда-ть 9МенюМС10Выход

```

Рис. 2.14: Изменяем файл

Создаём файл и запускаем файл

```

khokhlachevapolina@fedora:~/work/arch-pc/lab08$ nasm -f elf lab8-3.asm
khokhlachevapolina@fedora:~/work/arch-pc/lab08$ ld -m elf_i386 -o lab8-3 lab8-3.o
khokhlachevapolina@fedora:~/work/arch-pc/lab08$ ./lab8-3 5 3 4
Результат:0
khokhlachevapolina@fedora:~/work/arch-pc/lab08$

```

Рис. 2.15: Проверяем работу

Задание для самостоятельной работы

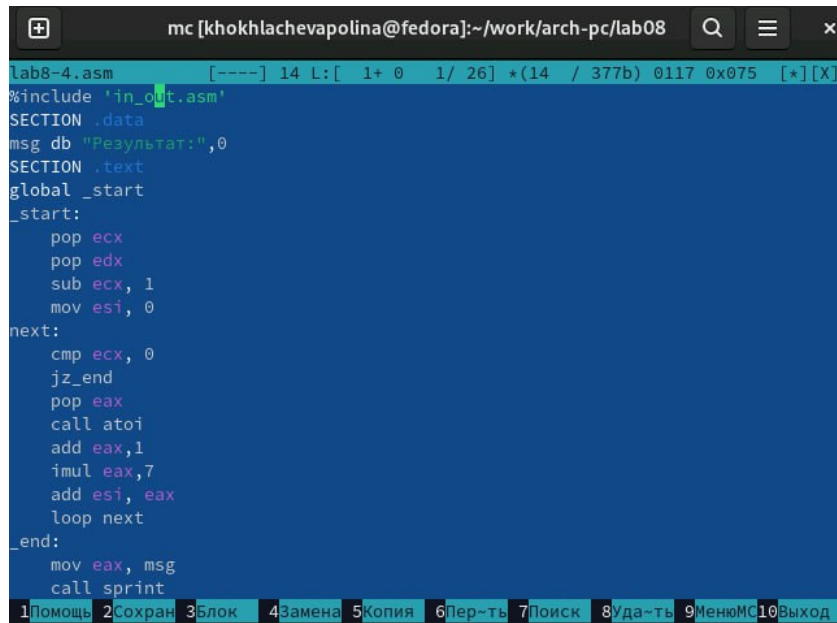
1. Напишите программу, которая находит сумму значений функции $f(x)$ для $x = 1, 2, \dots, n$, т.е. программа должна выводить значение $f(1) + f(2) + \dots + f(n)$. Значения n передаются как аргументы. Вид функции $f(x)$ выбрать из таблицы 8.1 вариантов заданий в соответствии с вариантом, полученным при выполнении лабораторной работы № 7. Создайте исполняемый файл и проверьте его работу на нескольких наборах $x = 1, 2, \dots, n$.

7(1 + 1) Создаём новый файл

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

Рис. 2.16: Создаём файл командой touch

Пишем программу для решения выражения



```
lab8-4.asm  [----] 14 L: [ 1+ 0 1/ 26] *(14 / 377b) 0117 0x075 [*][X]
#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, 0
    jz _end
    pop eax
    call atoi
    add eax, 1
    imul eax, 7
    add esi, eax
    loop next
_end:
    mov eax, msg
    call sprint
```

1Помощь 2Сохран 3Блок 4Замена 5Копия 6Пер-ть 7Поиск 8Уда-ть 9МенюMC 10Выход

Рис. 2.17: Создаём файл командой touch

3 Выводы

Мы научились решать программы с использованием циклов и обработкой аргументов командной строки