

Отчёт по лабораторной работе №9

Компьютерные науки и технология программирования

Сячинова Ксения Ивановна

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Выводы	16

Список иллюстраций

2.1	Создание каталога и файла	6
2.2	Текст программы	7
2.3	Проверка программы	7
2.4	Изменение программы	8
2.5	Результат программы	9
2.6	Изменение программы	9
2.7	Результата программы	10
2.8	Создание файла	10
2.9	Текст программы	10
2.10	Результат программы	11
2.11	Создание файла	11
2.12	Текст программы	12
2.13	Проверка работы программы	12
2.14	Изменение программы	13
2.15	Проверка работы программы	13
2.16	Текст программы	14
2.17	Проверка программы	15

Список таблиц

1 Цель работы

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

2 Выполнение лабораторной работы

1. Создаём каталог для программ лабораторных №9, переходим в него и создаём файл 'lab9-1.asm'. Открываем его.(рис. 2.1)

```
kisyachinova1@dk8n73 ~ $ cd work
kisyachinova1@dk8n73 ~/work $ cd arch-pc
kisyachinova1@dk8n73 ~/work/arch-pc $ mkdir lab09
kisyachinova1@dk8n73 ~/work/arch-pc $ cd lab09
kisyachinova1@dk8n73 ~/work/arch-pc/lab09 $ touch lab9-1.asm
kisyachinova1@dk8n73 ~/work/arch-pc/lab09 $ mcedit lab9-1.asm
```

Рис. 2.1: Создание каталога и файла

Вводим текст программы. (рис. 2.2),

```

lab9-1.asm [----] 9 L:
#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

```

Рис. 2.2: Текст программы

Создаём исполняемый файл и проверяем работу. Я проверила работу для N=3, N=4. (рис. 2.3)

```

kisyachinova1@dk8n73 ~/work/arch-pc/lab09 $ nasm -f elf lab9-1.asm
kisyachinova1@dk8n73 ~/work/arch-pc/lab09 $ ld -m elf_i386 -o lab9-1 lab9-1.o
kisyachinova1@dk8n73 ~/work/arch-pc/lab09 $ ./lab9-1
Введите N: 3
3
2
1
kisyachinova1@dk8n73 ~/work/arch-pc/lab09 $ ./lab9-1
Введите N: 4
4
3
2
1
kisyachinova1@dk8n73 ~/work/arch-pc/lab09 $

```

Рис. 2.3: Проверка программы

Изменим значение регистра в 'ecx' в цикле.(рис. 2.4)

```
lab9-1.asm [----] 4 L: 1+2
#include 'io_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

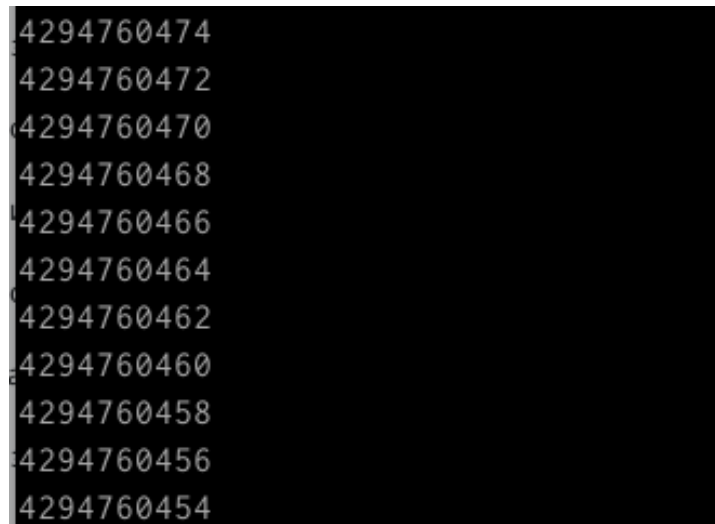
    mov eax,N
    call atoi
    mov [N],eax

    mov ecx,[N]
label:
    sub ecx,1
    mov [N],ecx
    mov eax,[N]
    call iprintLF
    loop label

call quit
```

Рис. 2.4: Изменение программы

Данный пример показывает, что использование регистра 'ecx' в теле цикла 'loop' может привести к некорректной работе программы.(рис. 2.5)

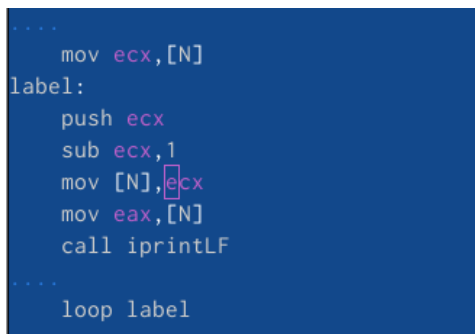


4294760474
4294760472
4294760470
4294760468
4294760466
4294760464
4294760462
4294760460
4294760458
4294760456
4294760454

Рис. 2.5: Результат программы

Программа работает некорректно. Регистр 'ecx' принимает бесконечное количество значений. Число проходов цикла не соответствует числу N.

Для использования регистра 'ecx' в цикле и сохранения корректности работы использует стек. Вносим изменения в программу, где добавим команды 'push' и 'pop'. (рис. 2.6)



```
.....  
    mov ecx,[N]  
label:  
    push ecx  
    sub ecx,1  
    mov [N],ecx  
    mov eax,[N]  
    call iprintLF  
.....  
    loop label
```

Рис. 2.6: Изменение программы

Файл работает корректно, в данном случае число проходов цикла соответствует значению N.(рис. 2.7)

```

kisyachinova1@dk8n73 ~/work/arch-pc/lab09 $ nasm -f elf lab9-1.asm
kisyachinova1@dk8n73 ~/work/arch-pc/lab09 $ ld -m elf_i386 -o lab9-1 lab9-1.o
kisyachinova1@dk8n73 ~/work/arch-pc/lab09 $ ./lab9-1
Введите N: 3
2
1
0
kisyachinova1@dk8n73 ~/work/arch-pc/lab09 $ ./lab9-1
Введите N: 4
3
2
1
0

```

Рис. 2.7: Результаты программы

2. Создадим файл 'lab9-2.asm', откроем его и введём текст программы. (рис. 2.8), (рис. 2.9)

```

kisyachinova1@dk8n73 ~/work/arch-pc/lab09 $ touch lab9-2.asm
kisyachinova1@dk8n73 ~/work/arch-pc/lab09 $ mcedit lab9-2.asm

```

Рис. 2.8: Создание файла

```

%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 sprintfLF
    loop next
    ....
_end:
    call quit

```

Рис. 2.9: Текст программы

В результате все введённые аргументы обрабатываются. (рис. 2.10)

```

kisyachinova1@dk8n73 ~/work/arch-pc/lab09 $ nasm -f elf lab9-2.asm
kisyachinova1@dk8n73 ~/work/arch-pc/lab09 $ ld -m elf_i386 -o lab9-2 lab9-2.o
kisyachinova1@dk8n73 ~/work/arch-pc/lab09 $ ./lab9-2 1 2 3
1
2
3
kisyachinova1@dk8n73 ~/work/arch-pc/lab09 $ ./lab9-2 1 2 3 4
1
2
3
4

```

Рис. 2.10: Результат программы

Рассмотрим программу, которая выводит сумму чисел. Для этого создадим новый файл 'lab9-3.asm', откроем его и введём текст программы.(рис. 2.11), (рис. 2.12)

```

kisyachinova1@dk8n73 ~/work/arch-pc/lab09 $ touch lab9-3.asm
kisyachinova1@dk8n73 ~/work/arch-pc/lab09 $ mcedit lab9-3.asm

```

Рис. 2.11: Создание файла

```

lab9-3.asm      [----]  0 L:
#include 'in_out.asm'

SECTION .data
msg db "Результат: ",0h
[
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: Текст программы

Программа работает корректно. (рис. 2.13)

```

kisyachinova1@dk8n73 ~/work/arch-pc/lab09 $ nasm -f elf lab9-3.asm
kisyachinova1@dk8n73 ~/work/arch-pc/lab09 $ ld -m elf_i386 -o lab9-3 lab9-3.o
kisyachinova1@dk8n73 ~/work/arch-pc/lab09 $ ./lab9-3 12 13 7 10 5
Результат: 47
kisyachinova1@dk8n73 ~/work/arch-pc/lab09 $ [

```

Рис. 2.13: Проверка работы программы

Изменим программу так, чтобы программа выводила произведение чисел, которые передаются в программу как аргументы. (рис. 2.14), (рис. 2.15)

```
next:
    cmp ecx, 0h
    jz _end
    . . . .
    pop eax
    call atoi
    mul esi
    mov esi, eax
    . . . .
    loop next
    . . . .
```

Рис. 2.14: Изменение программы

```
kisyachinova1@dk8n73 ~/work/arch-pc/lab09 $ nasm -f elf lab9-3.asm
kisyachinova1@dk8n73 ~/work/arch-pc/lab09 $ ld -m elf_i386 -o lab9-3 lab9-3.o
kisyachinova1@dk8n73 ~/work/arch-pc/lab09 $ ./lab9-3 1 3 9
Результат: 27
kisyachinova1@dk8n73 ~/work/arch-pc/lab09 $ ./lab9-3 1 4 25
Результат: 100
```

Рис. 2.15: Проверка работы программы

#Самостоятельная работа

Напишем программу, которая находит сумму значений функций $f(x)$ для $x=x_1, x_2, \dots, x_n$. Программа будет выводить значение $f(x_1)+f(x_2)+\dots+f(x_n)$. В соответствии с вариантом 11 имеем $f(x)=15x-9$. (рис. 2.16)

```

lab9-1.1.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, 0
    ....
next:
    cmp ecx,0h
    jz _end
    ....
    pop eax
    call atoi
    mov eax, eax
    mov ebx, 15
    mul ebx
    add eax, 2
    add esi,eax
    ....
    loop next
    ....
_end:
    mov eax,msg
    call sprint
    mov eax, esi
    call iprintLF
    call quit

```

Рис. 2.16: Текст программы

Программа работает корректно при заданных значениях. (рис. 2.17)

```
kisyachinova1@dk8n73 ~/work/arch-pc/lab09 $ nasm -f elf lab9-1.1.asm
kisyachinova1@dk8n73 ~/work/arch-pc/lab09 $ ld -m elf_i386 -o lab9-1.1 lab9-1.1.o
kisyachinova1@dk8n73 ~/work/arch-pc/lab09 $ ./lab9-1.1 1 2
Результат: 49
kisyachinova1@dk8n73 ~/work/arch-pc/lab09 $ ./lab9-1.1 2 5 1
Результат: 126
kisyachinova1@dk8n73 ~/work/arch-pc/lab09 $
```

Рис. 2.17: Проверка программы

3 Выводы

В ходе выполнения данной лабораторной работы я приобрела навыки написания программ с использованием циклов и обработкой аргументов командной строки