

# **Лабораторная работа №9**

**Программирование цикла. Обработка аргументов командной строки.**

Киньябаева Аиша Иделевна

# Содержание

1	Цель работы	4
2	Выполнение лабораторной работы	5
3	Выводы	13

## Список иллюстраций

2.1	код . . . . .	5
2.2	lab9-1.asm_вывод . . . . .	6
2.3	код . . . . .	6
2.4	lab9-1.asm_вывод . . . . .	7
2.5	lab9-1.asm_вывод . . . . .	7
2.6	код . . . . .	8
2.7	lab9-1.asm_вывод . . . . .	8
2.8	код . . . . .	9
2.9	вывод lab9-2.asm . . . . .	9
2.10	код . . . . .	10
2.11	вывод lab9-3.asm . . . . .	10
2.12	код . . . . .	11
2.13	вывод lab9-3.asm . . . . .	11
2.14	код . . . . .	12
2.15	2(x-1) вывод . . . . .	12

# 1 Цель работы

Целью данной работы является приобретение навыков написания программ с использованием циклов и обработкой аргументов командной строки. # Задание

Освоение команд написания циклов, работа с аргументами командной строки, написание программ с использованием аргументов.

## 2 Выполнение лабораторной работы

Написание первой программы lab9-1.asm, которая вычитает единицу из введенного значения, пока ecx не достигнет 0 (однако первым выводом является число 3, т.к. мы сопоставляем ecx=N) (рис. 2.1), (рис. 2.2)

```
%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. 2.1: код

```

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

```

Рис. 2.2: lab9-1.asm\_вывод

Далее мы преобразовываем программу, в результате чего она работает некорректно, отнимает две единицы вместо одной, а при введении нечетного числа выводит абсолютно не то, что нам нужно (рис. [fig:fig3]), (рис. 2.4), (рис. 2.5)

```

#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

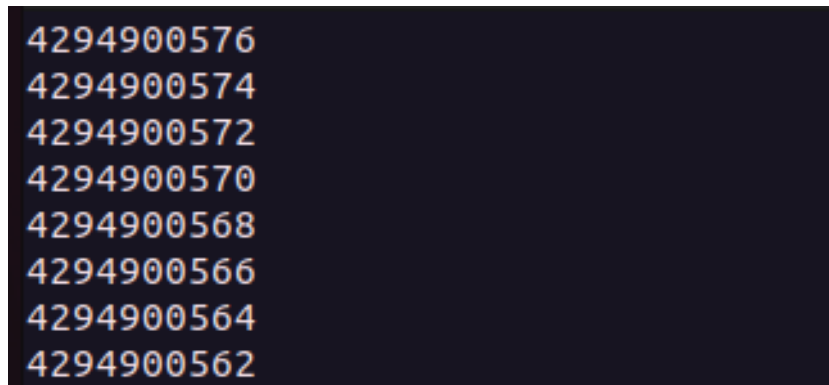
mov ecx, [N]

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

call quit
~

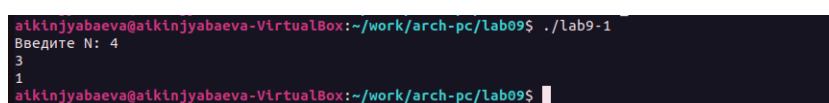
```

Рис. 2.3: код



```
4294900576
4294900574
4294900572
4294900570
4294900568
4294900566
4294900564
4294900562
```

Рис. 2.4: lab9-1.asm\_вывод



```
aikinjyabaeva@aikinjyabaeva-VirtualBox:~/work/arch-pc/lab09$ ./lab9-1
Введите N: 4
3
1
aikinjyabaeva@aikinjyabaeva-VirtualBox:~/work/arch-pc/lab09$
```

Рис. 2.5: lab9-1.asm\_вывод

Еще одно преобразование, в котором уже используется стек. А в результате мы получаем схожий с первой программой результат, но с выводом первого элемента уже уменьшенным на единицу (рис. 2.6), (рис. 2.7)

```

#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

call quit
~

```

Рис. 2.6: код

```

aikinjyabaeva@aikinjyabaeva-VirtualBox:~/work/arch-pc/lab09$ ld -m elf_i386 -o lab9-1 lab9-1.o
aikinjyabaeva@aikinjyabaeva-VirtualBox:~/work/arch-pc/lab09$ ./lab9-1
Введите N: 3
2
1
0
aikinjyabaeva@aikinjyabaeva-VirtualBox:~/work/arch-pc/lab09$

```

Рис. 2.7: lab9-1.asm\_вывод

Создаем новый файл lab9-2.asm, который выводит введенные аргументы. Видим как аргументом считается все до пробела (рис. 2.8), (рис. 2.9)



```

#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
~

```

Рис. 2.8: код

```

a1kinjyabaeva@a1kinjyabaeva-VirtualBox:~/work/arch-pc/lab09$ ./lab9-2 аргумент1 аргумент 2 'аргумент
3'
аргумент1
аргумент
2
аргумент 3
a1kinjyabaeva@a1kinjyabaeva-VirtualBox:~/work/arch-pc/lab09$

```

Рис. 2.9: вывод lab9-2.asm

Следующая программа lab9-3.asm. Она выводит сумму введенных аргументов (рис. 2.10), (рис. 2.11)

```

#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. 2.10: код

```

aikinjayabaeva@aikinjayabaeva-VirtualBox:~/work/arch-pc/lab09$ ld -m elf_i386 -o lab9-3 lab9-3.o
aikinjayabaeva@aikinjayabaeva-VirtualBox:~/work/arch-pc/lab09$ ./lab9-3 12 13 7 10 5
Результат: 47
aikinjayabaeva@aikinjayabaeva-VirtualBox:~/work/arch-pc/lab09$

```

Рис. 2.11: вывод lab9-3.asm

Преобразовываю программу так, чтобы она выводила произведение введенных аргументов (рис. 2.12), (рис. 2.13)

```

#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. 2.12: код

```

aikinjyabaeva@aikinjyabaeva-VirtualBox:~/work/arch-pc/lab09$ nasm -f elf lab9-3.asm
aikinjyabaeva@aikinjyabaeva-VirtualBox:~/work/arch-pc/lab09$ ld -m elf_i386 -o lab9-3 lab9-3.o
aikinjyabaeva@aikinjyabaeva-VirtualBox:~/work/arch-pc/lab09$ ./lab9-3 3 7 8
Результат: 168
aikinjyabaeva@aikinjyabaeva-VirtualBox:~/work/arch-pc/lab09$ █

```

Рис. 2.13: вывод lab9-3.asm

## САМОСТОЯТЕЛЬНАЯ РАБОТА

Из прошлой лабораторной работы мне попался 4й вариант.

Пишу программу, выводящую заданную формулу и сумму вычисленных значений (рис. 2.14), (рис. 2.15)

```

#include 'in_out.asm'

SECTION .data
msg1 db "f(x)=2(x-1)",0
msg2 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
dec eax

mov ebx,2
mul ebx
add esi,eax
loop next

_end:
mov eax,msg1
call sprintf
mov eax,msg2
call sprintf
mov eax,esi
call sprintf

call quit
~

```

Рис. 2.14: код

```

a1kinjyabaeva@a1kinjyabaeva-VirtualBox:~/work/arch-pc/lab09$ ld -m elf_i386 -o mywork mywork.o
a1kinjyabaeva@a1kinjyabaeva-VirtualBox:~/work/arch-pc/lab09$ ./mywork 3 4 5
f(x)=2(x-1)
Результат: 18
a1kinjyabaeva@a1kinjyabaeva-VirtualBox:~/work/arch-pc/lab09$ ./mywork 4 7 11 2
f(x)=2(x-1)
Результат: 40

```

Рис. 2.15:  $2(x-1)$  вывод

Загрузка всех файлов на Git.

Далее создается отчет по 9й лабораторной работе с помощью Markdown.

## **3 Выводы**

В ходе данной лабораторной работы были изучены циклы и обработка аргументов командной строки. Были освоены и использованы на практике основные команды циклов и стеков.