

Отчет по лабораторной работе №9

Компьютерные науки и технология программирования

Попова Елизавета Сергеевна

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Самостоятельная работа	16
4	Выводы	18

Список иллюстраций

2.1	Создание каталога и файла	6
2.2	Программа lab9-1.asm	7
2.3	Проверка работы lab9-1.asm	8
2.4	Изменение программы lab9-1.asm	9
2.5	Измененная программа lab9-1.asm	10
2.6	Дважды измененная программа lab9-1.asm	11
2.7	Работа измененной программы lab9-1.asm	12
2.8	Создание файла lab9-2.asm	12
2.9	Текст программы lab9-2.asm	13
2.10	Проверка работы lab9-2.asm	13
2.11	Создание файла lab9-3.asm	14
2.12	Текст программы lab9-3.asm	14
2.13	Проверка программы lab9-3.asm	14
2.14	Изменение программы lab9-3.asm	15
2.15	Проверка работы измененной программы lab9-3.asm	15
3.1	Текст программы lab9-4.asm	16
3.2	Проверка программы lab9-4.asm	17

Список таблиц

1 Цель работы

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


2 Выполнение лабораторной работы

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

```
[laisacreate@10 ~]$ mkdir work/arch-pc/lab09
[laisacreate@10 ~]$ cd ~work/arch-pc/lab09
bash: cd: ~work/arch-pc/lab09: Нет такого файла или каталога
[laisacreate@10 ~]$ cd work/arch-pc/lab09
[laisacreate@10 lab09]$ touch lab9-1.asm
[laisacreate@10 lab09]$
```

Рис. 2.1: Создание каталога и файла

Вводим текст программы. (рис. 2.2)

Открыть ▾  • lab9-1.asm
~/work/arch-pc/lab09

```
Листинг 9.1. Программа вывода значений регистра ecx
;-----
; Программа вывода значений регистра 'ecx'
;-----
%include 'in_out.asm'
SECTION .data
msg1 db 'Введите N: ',0h
SECTION .bss
N: resb 10
SECTION .text
global _start
_start:
; ---- Вывод сообщения 'Введите N: '
mov eax,msg1
call sprint
; ---- Ввод 'N'
mov ecx, N
mov edx, 10
call sread
; ---- Преобразование 'N' из символа в число
mov eax,N
call atoi
mov [N],eax
; ----- Организация цикла
mov ecx,[N] ; Счетчик цикла, `ecx=N`
label:
mov [N],ecx
mov eax,[N]
call iprintLF ; Вывод значения `N`
loop label ; `ecx=ecx-1` и если `ecx` не '0'
; переход на `label`
call quit
```

Рис. 2.2: Программа lab9-1.asm

Создаём исполняемый файл и проверяем работу. (рис. 2.3)

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

Рис. 2.3: Проверка работы lab9-1.asm

Изменим текст программы lab9-1.asm. (рис. 2.4)


```

; Программа вывода значений регистра 'ecx'

%include 'in_out.asm'

SECTION .data
msg1 db 'Введите N: ',0h

SECTION .bss
N: resb 10

SECTION .text
global _start
_start:

; ----- Вывод сообщения 'Введите N: '
mov eax,msg1
call sprint

; ----- Ввод 'N'
mov ecx, N
mov edx, 10
call sread

; ----- Преобразование 'N' из символа в число
mov eax,N
call atoi
mov [N],eax

; ----- Организация цикла
mov ecx,[N] ; Счетчик цикла, `ecx=N`
label:
sub ecx,1
mov [N],ecx
mov eax,[N]
call iprintLF ; Вывод значения `N`
loop label ;
; переход на `label`
call quit

```


Рис. 2.4: Изменение программы lab9-1.asm

Видим, что данный алгоритм выводит нечетные числа, которые меньше, чем N. Получим всего $N/2$ значений.

```
[laisacrete@10 lab09]$ nasm -f elf lab9-1.asm
[laisacrete@10 lab09]$ ld -m elf_i386 -o lab9-1 lab9-1.o
[laisacrete@10 lab09]$ ./lab9-1
Введите N: 10
9
7
5
3
1
```

Рис. 2.5: Измененная программа lab9-1.asm

Вновь изменим код программы lab9-1.asm. (рис. 2.6)

Открыть ▾ 

• lab9-1.asm
~/work/arch-pc/lab09

```
; Программа вывода значений регистра 'ecx'

%include 'in_out.asm'

SECTION .data
msg1 db 'Введите N: ',0h

SECTION .bss
N: resb 10

SECTION .text
global _start
_start:

; ----- Вывод сообщения 'Введите N: '
mov eax,msg1
call sprint

; ----- Ввод 'N'
mov ecx, N
mov edx, 10
call sread

; ----- Преобразование 'N' из символа в число
mov eax,N
call atoi
mov [N],eax

; ----- Организация цикла
mov ecx,[N] ; Счетчик цикла, `ecx=N`
label:
push ecx
sub ecx,1
mov [N],ecx
mov eax,[N]
call iprintLF ; Вывод значения `N`
pop ecx
loop label ;
; переход на `label`
call quit
```

Рис. 2.6: Дважды измененная программа lab9-1.asm

Проверим работу файла. Он работает верно. Всего выведено N значений. (рис. 2.7)

```

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

```

Рис. 2.7: Работа измененной программы lab9-1.asm

Создадим файл 'lab9-2.asm'. (рис. 2.8)

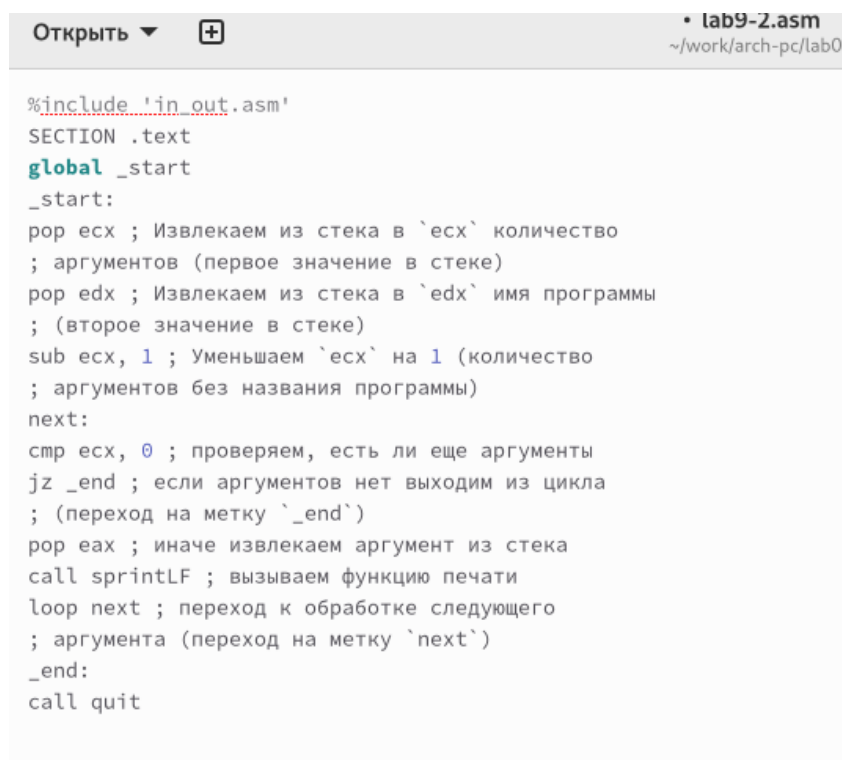
```

0
[laisacrete@10 lab09]$ touch lab9-2.asm

```

Рис. 2.8: Создание файла lab9-2.asm

Введем нужный текст в файл lab9-2.asm. (рис. 2.9)




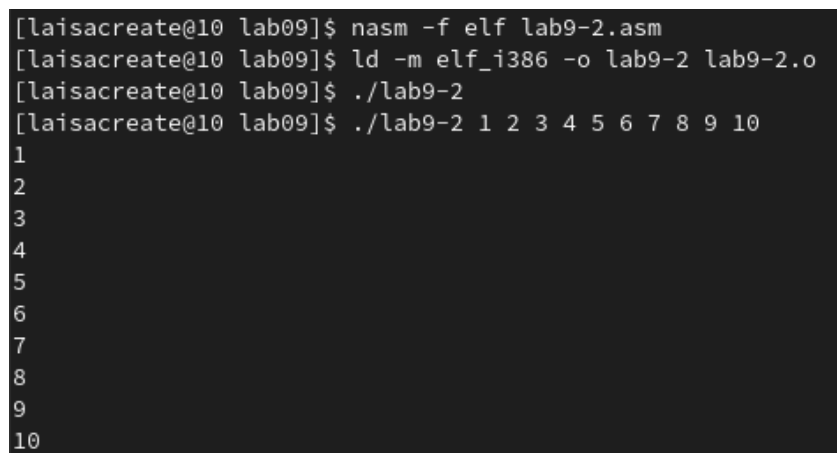
```
Открыть ▾  • lab9-2.asm  
~/work/arch-pc/lab0  
  
%include 'in_out.asm'  
SECTION .text  
global _start  
_start:  
    pop ecx ; Извлекаем из стека в `ecx` количество  
            ; аргументов (первое значение в стеке)  
    pop edx ; Извлекаем из стека в `edx` имя программы  
            ; (второе значение в стеке)  
    sub ecx, 1 ; Уменьшаем `ecx` на 1 (количество  
            ; аргументов без названия программы)  
next:  
    cmp ecx, 0 ; проверяем, есть ли еще аргументы  
    jz _end ; если аргументов нет выходим из цикла  
            ; (переход на метку `_end`)  
    pop eax ; иначе извлекаем аргумент из стека  
    call printf ; вызываем функцию печати  
    loop next ; переход к обработке следующего  
            ; аргумента (переход на метку `next`)  
_end:  
    call quit
```

Рис. 2.9: Текст программы lab9-2.asm

Запустим программу lab9-2.asm и проверим работу. (рис. 2.10)



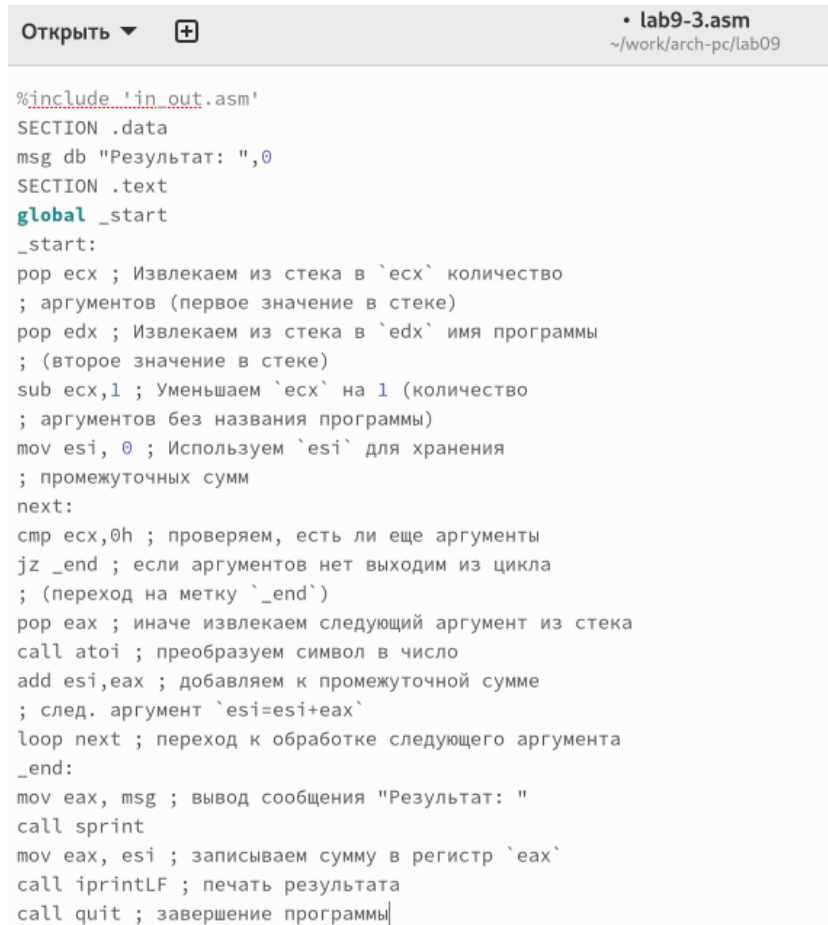
```
[laisacre@10 lab09]$ nasm -f elf lab9-2.asm  
[laisacre@10 lab09]$ ld -m elf_i386 -o lab9-2 lab9-2.o  
[laisacre@10 lab09]$ ./lab9-2  
[laisacre@10 lab09]$ ./lab9-2 1 2 3 4 5 6 7 8 9 10  
1  
2  
3  
4  
5  
6  
7  
8  
9  
10
```

Рис. 2.10: Проверка работы lab9-2.asm

Создадим файл lab9-3.asm и введем текст программы. (рис. 2.11), (рис. 2.12)

```
[laisacrete@10 lab09]$ touch lab9-3.asm
```

Рис. 2.11: Создание файла lab9-3.asm



```
• lab9-3.asm
~/.work/arch-pc/lab09

%include 'in_out.asm'
SECTION .data
msg db "Результат: ",0
SECTION .text
global _start
_start:
pop ecx ; Извлекаем из стека в `ecx` количество
; аргументов (первое значение в стеке)
pop edx ; Извлекаем из стека в `edx` имя программы
; (второе значение в стеке)
sub ecx,1 ; Уменьшаем `ecx` на 1 (количество
; аргументов без названия программы)
mov esi, 0 ; Используем `esi` для хранения
; промежуточных сумм
next:
cmp ecx,0h ; проверяем, есть ли еще аргументы
jz _end ; если аргументов нет выходим из цикла
; (переход на метку `_end`)
pop eax ; иначе извлекаем следующий аргумент из стека
call atoi ; преобразуем символ в число
add esi,eax ; добавляем к промежуточной сумме
; след. аргумент `esi=esi+eax`
loop next ; переход к обработке следующего аргумента
_end:
mov eax, msg ; вывод сообщения "Результат: "
call sprint
mov eax, esi ; записываем сумму в регистр `eax`
call iprintLF ; печать результата
call quit ; завершение программы
```

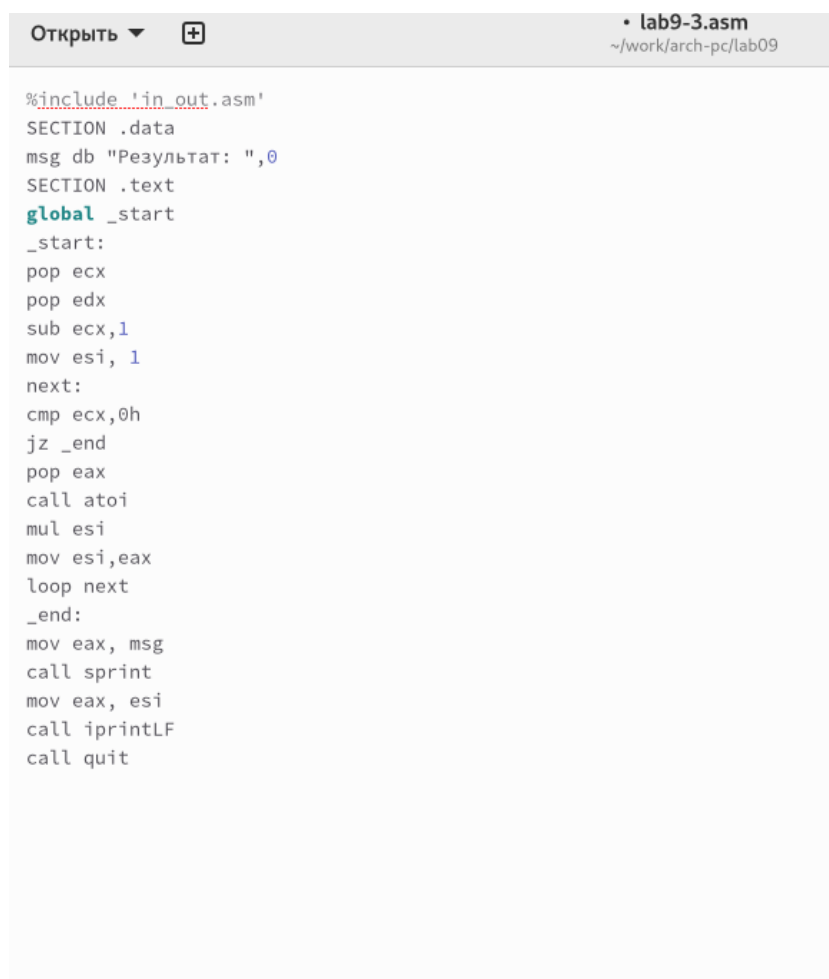
Рис. 2.12: Текст программы lab9-3.asm

Проверим работу файла. Программа выдает верный результат. (рис. 2.13)

```
[laisacrete@10 lab09]$ nasm -f elf lab9-3.asm
[laisacrete@10 lab09]$ ld -m elf_i386 -o lab9-3 lab9-3.o
[laisacrete@10 lab09]$ ./lab9-3 10 20 30 40 50
Результат: 150
```

Рис. 2.13: Проверка программы lab9-3.asm

Изменим программу так, чтобы выводилось произведение чисел. (рис. 2.14)

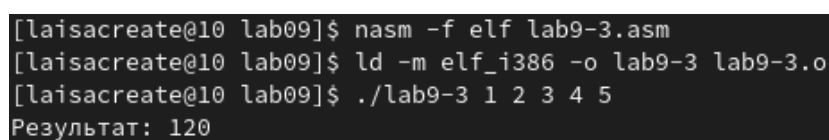


```
Открыть ▾ + • lab9-3.asm
~/work/arch-pc/lab09

%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
```

Рис. 2.14: Изменение программы lab9-3.asm

Проверим работу программы lab9-3.asm. (рис. 2.15)



```
[laisacrete@10 lab09]$ nasm -f elf lab9-3.asm
[laisacrete@10 lab09]$ ld -m elf_i386 -o lab9-3 lab9-3.o
[laisacrete@10 lab09]$ ./lab9-3 1 2 3 4 5
Результат: 120
```

Рис. 2.15: Проверка работы измененной программы lab9-3.asm

3 Самостоятельная работа

Напишем программу, которая находит сумму значений функций $f(x)$ для $x=x_1, x_2, \dots, x_n$. Программа будет выводить значение $f(x_1)+f(x_2)+\dots+f(x_n)$. Моя функция $f(x) = 6x + 13$. (рис. 2.16) (рис. 3.1)




```
Открыть ▾  • lab9-4.asm  
~/work/arch-pc/lab09  
  
%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,6  
mul ebx  
add eax,13  
add esi,eax  
loop next  
_end:  
mov eax,msg  
call sprint  
mov eax,esi  
call iprintLF  
call quit
```

Рис. 3.1: Текст программы lab9-4.asm

Программа работает верно и выдает верное значение, то есть задание выполнено верно. (рис. 3.2)

```
[laisacreate@10 lab09]$ nasm -f elf lab9-4.asm
lab9-4.asm:20: warning: label alone on a line without a command
[-w+label-orphan]
[laisacreate@10 lab09]$ ld -m elf_i386 -o lab9-4 lab9-4.o
[laisacreate@10 lab09]$ ./lab9-4 1 2 3
Результат: 75
[laisacreate@10 lab09]$ ./lab9-4 5 5 5
Результат: 129
```

Рис. 3.2: Проверка программы lab9-4.asm

4 Выводы

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