

Лабораторная работа №9

Архитектура компьютеров

Скандарова Полина Юрьевна

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Задание для самостоятельной работы	12
4	Выводы	13

Список иллюстраций

2.1	Текст листинга в файле.	6
2.2	Проверка работы программы.	7
2.3	Изменение текста листинга в файле.	7
2.4	Изменение текста листинга в файле.	8
2.5	Проверка работы программы.	8
2.6	Текст листинга в файле.	9
2.7	Проверка работы программы.	9
2.8	Текст листинга в файле.	10
2.9	Проверка работы программы.	10
2.10	Изменение текста листинга в файле.	11
2.11	Проверка работы программы.	11
3.1	Программа в файле.	12
3.2	Проверка работы программы.	12

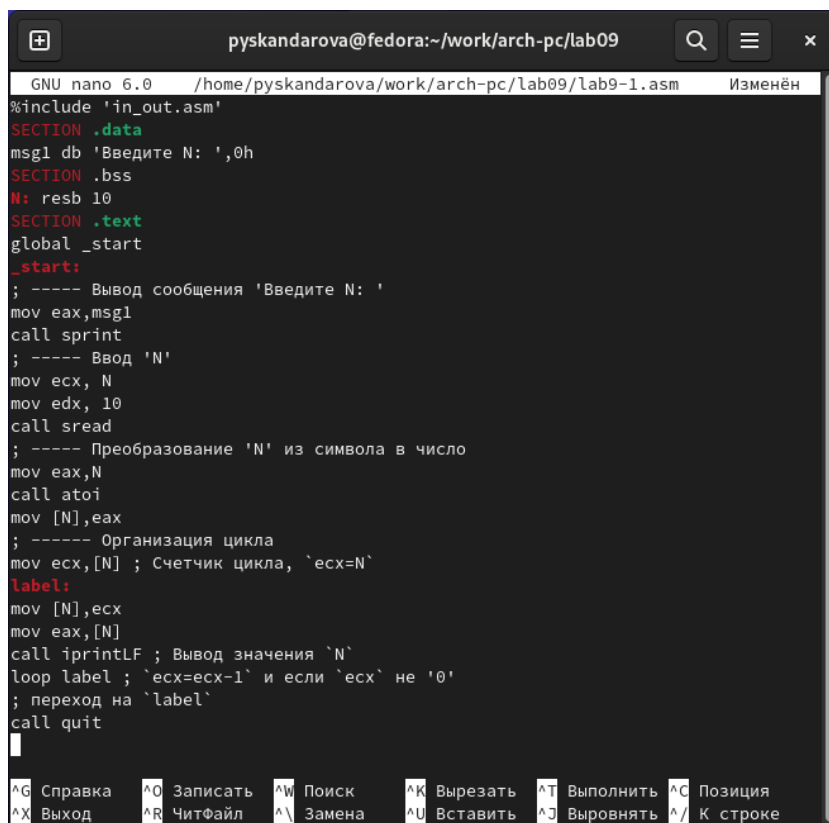
Список таблиц

1 Цель работы

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

2 Выполнение лабораторной работы

Для начала необходимо создать каталог для программ лабораторной работы №9, перейти в него и создать файл lab9-1.asm. Потом нужно внимательно изучить текст программы в листинге и перенести его в файл. (рис. 2.1)



```
GNU nano 6.0 /home/pyskandarova/work/arch-pc/lab09/lab9-1.asm Изменён
%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
```

^G Справка ^O Записать ^W Поиск ^K Вырезать ^T Выполнить ^C Позиция
^X Выход ^R ЧитФайл ^\ Замена ^U Вставить ^J Выровнять ^_ К строке

Рис. 2.1: Текст листинга в файле.

Программа пишет числа от введенного до одного.(рис. 2.2)

```

[pyskandarova@fedora lab09]$ nasm -f elf lab9-1.asm
[pyskandarova@fedora lab09]$ ld -m elf_i386 -o lab9-1 lab9-1.o
[pyskandarova@fedora lab09]$ ./lab9-1
Введите N: 5
5
4
3
2
1
[pyskandarova@fedora lab09]$ ./lab9-1
Введите N: 7
7
6
5
4
3
2
1

```

Рис. 2.2: Проверка работы программы.

Изменив программу, добавив изменение значения регистра `ecx` в цикле. (рис. 2.3)

```

label:
sub ecx,1
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. 2.3: Изменение текста листинга в файле.

Получаю огромное количество проходов цикла, не соответствующее введённому числу. Для использования регистра `ecx` в цикле и сохранения корректности работы вношу изменения в текст программы добавив команды `push` и `pop` (добавления в стек и извлечения из стека) для сохранения значения счетчика цикла `loop`. (рис. 2.4)

```

label:
push ecx
sub ecx,1
mov [N],ecx
mov eax,[N]
call iprintLF ; Вывод значения `N`
pop ecx
loop label ; `ecx=ecx-1` и если `ecx` не `0`
; переход на `label`
call quit

```

Рис. 2.4: Изменение текста листинга в файле.

Теперь программа выводит числа от введённого минус один до нуля. Число проходов введённому числу соответствует. (рис. 2.5)

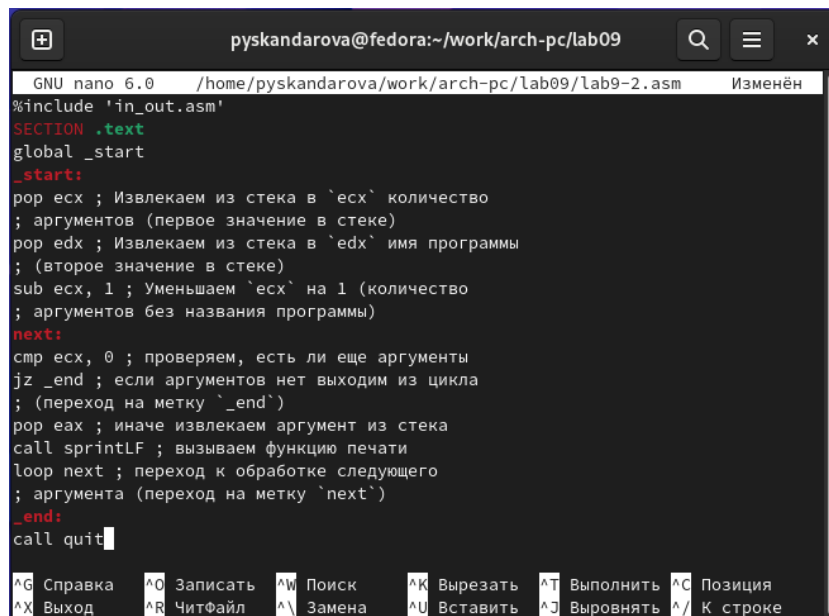
```

[pyskandarova@fedora lab09]$ nasm -f elf lab9-1.asm
[pyskandarova@fedora lab09]$ ld -m elf_i386 -o lab9-1 lab9-1.o
[pyskandarova@fedora lab09]$ ./lab9-1
Введите N: 5
4
3
2
1
0

```

Рис. 2.5: Проверка работы программы.

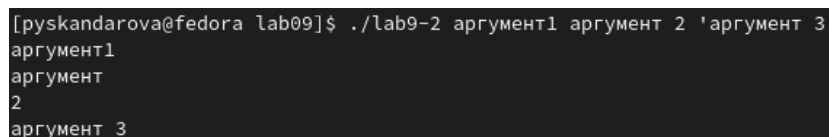
Теперь нужно внимательно изучить текст программы листинга и ввести его в файл lab9-2.asm. (рис. 2.6)



```
GNU nano 6.0 /home/pyskandarova/work/arch-pc/lab09/lab9-2.asm Изменён
%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 sprintf ; вызываем функцию печати
loop next ; переход к обработке следующего
; аргумента (переход на метку `next`)
_end:
call quit
```

Рис. 2.6: Текст листинга в файле.

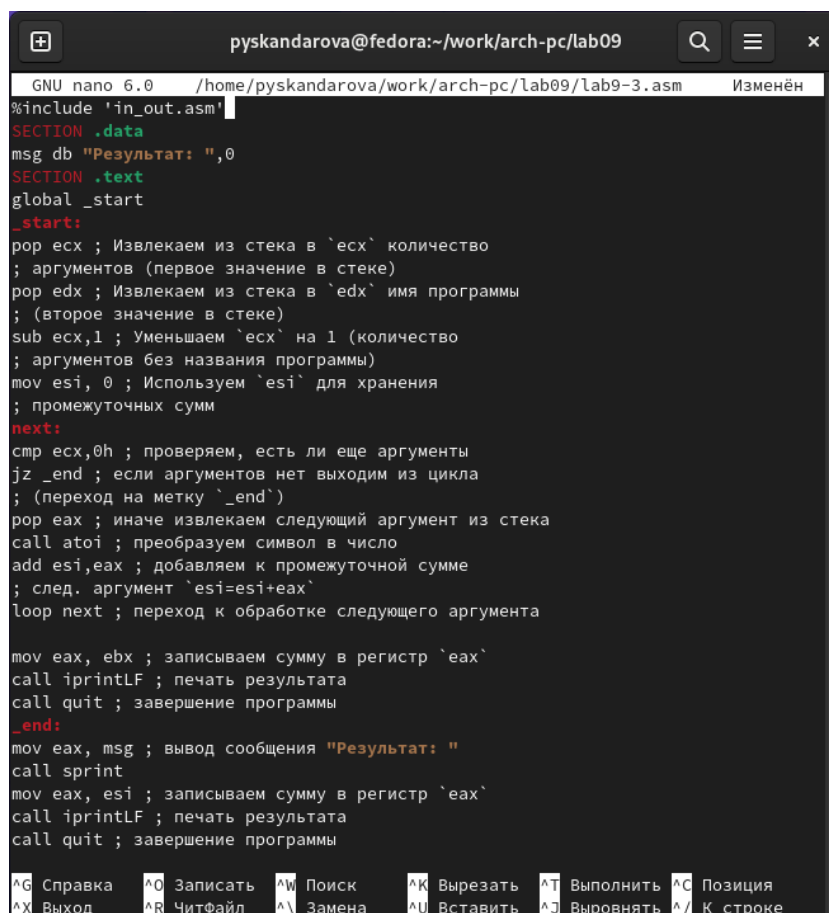
Создаю исполняемый файл и запускаю его, указав предложенные аргументы.
(рис. 2.7)



```
[pyskandarova@fedora lab09]$ ./lab9-2 аргумент1 аргумент 2 'аргумент 3'
аргумент1
аргумент
2
аргумент 3
```

Рис. 2.7: Проверка работы программы.

Обработано программой было четыре аргумента. Дальше создаю файл lab9-3.asm и ввожу в него текст программы из листинга. (рис. 2.8)



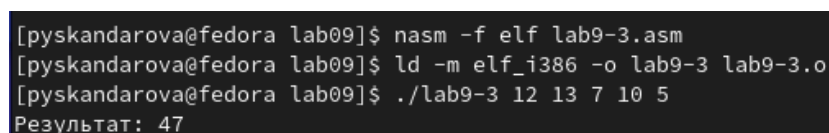
```
GNU nano 6.0 /home/pyskandarova/work/arch-pc/lab09/lab9-3.asm Изменён
%include 'in_out.asm'
SECTION .data
msg db "Результат: ",0
SECTION .text
global _start
_start:
pop ecx ; Извлекаем из стека в `ecx` количество
; аргументов (первое значение в стеке)
pop edx ; Извлекаем из стека в `edx` имя программы
; (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

mov eax, ebx ; записываем сумму в регистр `eax`
call iprintLF ; печать результата
call quit ; завершение программы
_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

^G Справка ^O Записать ^W Поиск ^K Вырезать ^T Выполнить ^C Позиция
^X Выход ^R ЧитФайл ^\ Замена ^U Вставить ^J Выровнять ^/_ К строке
```

Рис. 2.8: Текст листинга в файле.

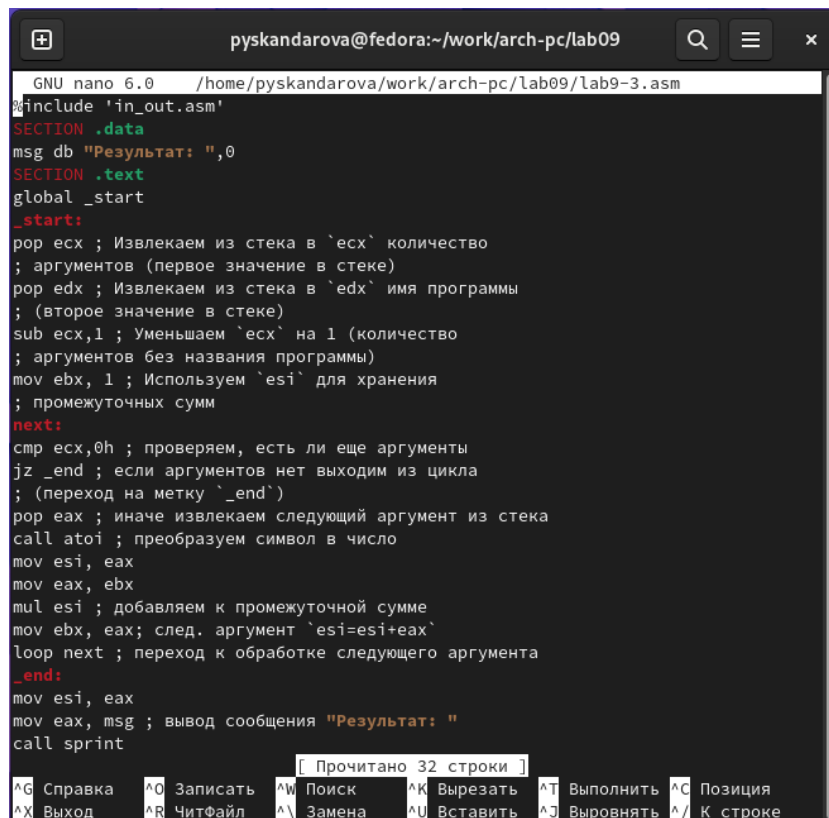
Создаю исполняемый файл и запускаю его, указав аргументы. (рис. 2.9)



```
[pyskandarova@fedora lab09]$ nasm -f elf lab9-3.asm
[pyskandarova@fedora lab09]$ ld -m elf_i386 -o lab9-3 lab9-3.o
[pyskandarova@fedora lab09]$ ./lab9-3 12 13 7 10 5
Результат: 47
```

Рис. 2.9: Проверка работы программы.

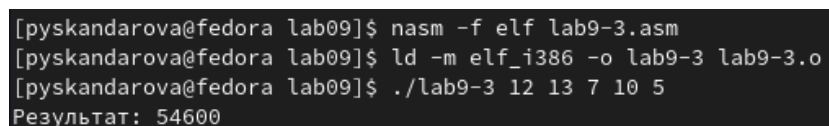
Изменяю текст программы из листинга для вычисления произведения аргументов командной строки.(рис. 2.10)



```
GNU nano 6.0 /home/pyskandarova/work/arch-pc/lab09/lab9-3.asm
%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 ebx, 1 ; Используем `esi` для хранения
; промежуточных сумм
next:
str 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
mov esi, eax
mov eax, ebx
mul esi ; добавляем к промежуточной сумме
mov ebx, eax; след. аргумент `esi=esi+eax`
loop next ; переход к обработке следующего аргумента
_end:
mov esi, eax
mov eax, msg ; вывод сообщения "Результат: "
call sprint
[ Прочитано 32 строки ]
^G Справка ^O Записать ^W Поиск ^K Вырезать ^T Выполнить ^C Позиция
^X Выход ^R ЧитФайл ^\ Замена ^U Вставить ^J Выводить ^/_ К строке
```

Рис. 2.10: Изменение текста листинга в файле.

И запускаю программу. (рис. 2.11)

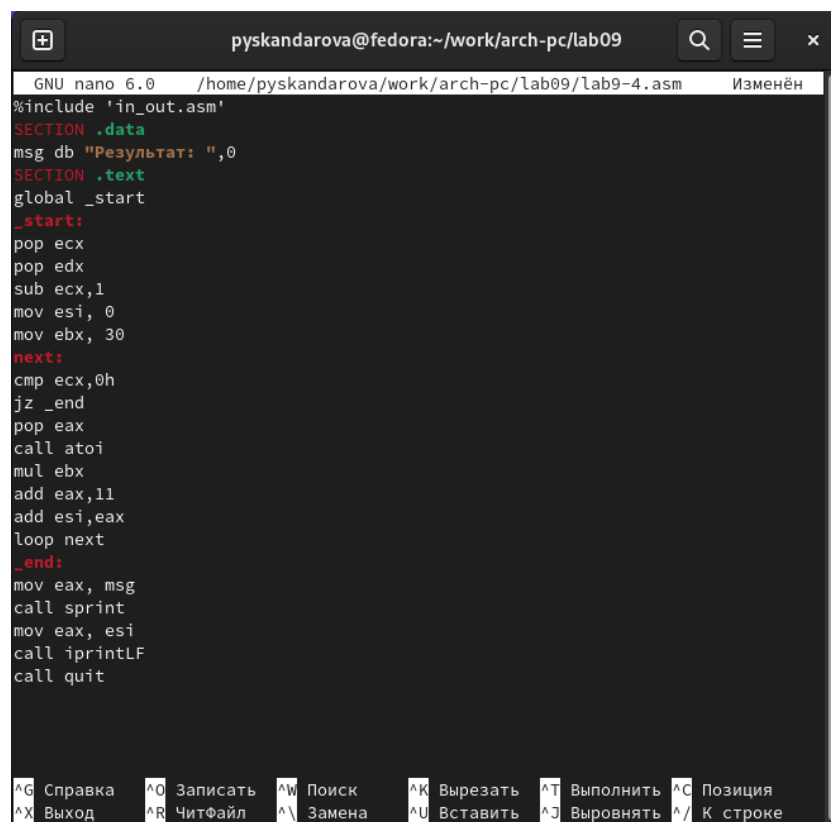


```
[pyskandarova@fedora lab09]$ nasm -f elf lab9-3.asm
[pyskandarova@fedora lab09]$ ld -m elf_i386 -o lab9-3 lab9-3.o
[pyskandarova@fedora lab09]$ ./lab9-3 12 13 7 10 5
Результат: 54600
```

Рис. 2.11: Проверка работы программы.

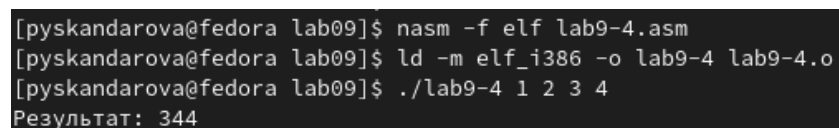
3 Задание для самостоятельной работы

Необходимо написать программу, которая находит сумму значений функции $30x-11$ для всех заданных аргументов.(рис. 3.1)(рис. 3.2)



```
GNU nano 6.0 /home/pyskandarova/work/arch-pc/lab09/lab9-4.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
mov ebx, 30
next:
cmp ecx,0h
jz _end
pop eax
call atoi
mul ebx
add eax,11
add esi,eax
loop next
_end:
mov eax, msg
call sprint
mov eax, esi
call iprintLF
call quit
```

Рис. 3.1: Программа в файле.



```
[pyskandarova@fedora lab09]$ nasm -f elf lab9-4.asm
[pyskandarova@fedora lab09]$ ld -m elf_i386 -o lab9-4 lab9-4.o
[pyskandarova@fedora lab09]$ ./lab9-4 1 2 3 4
Результат: 344
```

Рис. 3.2: Проверка работы программы.

4 Выводы

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