

# **Лабораторная работа №9**

**Архитектура компьютеров**

Кирилюк Светлана Алексеевна

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Выполнение лабораторной работы</b>	<b>6</b>
<b>3</b>	<b>Выводы</b>	<b>14</b>

## Список иллюстраций

2.1	Создание каталога и файла . . . . .	6
2.2	Ввод программы из листинга . . . . .	7
2.3	Создание исполняемого файла . . . . .	7
2.4	Изменение текста файла . . . . .	8
2.5	Создание исполняемого файла . . . . .	8
2.6	Изменение текста файла . . . . .	9
2.7	Исполнение файла . . . . .	9
2.8	Создание файла . . . . .	10
2.9	Ввод текста файла . . . . .	10
2.10	Создание исполняемого файла . . . . .	10
2.11	Создание нового файла . . . . .	10
2.12	Ввод текста файла . . . . .	11
2.13	Создание исполняемого файла . . . . .	11
2.14	Изменение текста файла . . . . .	12
2.15	Создание исполняемого файла . . . . .	12
2.16	Изменение текста файла . . . . .	13
2.17	Создание исполняемого файла . . . . .	13

## **Список таблиц**

# 1 Цель работы

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

## 2 Выполнение лабораторной работы

Создаём каталог для программ лабораторной работы №9, переходим в него и создаём файл lab9-1.asm (рис. 2.1).

```
sakirilyuk@dk5n60 ~ $ mkdir ~/work/arch-pc/lab09
sakirilyuk@dk5n60 ~ $ cd ~/work/arch-pc/lab09
sakirilyuk@dk5n60 ~/work/arch-pc/lab09 $ touch lab9-1.asm
sakirilyuk@dk5n60 ~/work/arch-pc/lab09 $
```



Рис. 2.1: Создание каталога и файла

Вводим в файл текст программы из 1-го листинга (рис. 2.2) и создаём исполняемый файл (рис. 2.3).

```

/afs/.dk.sci.pfu.edu.ru/home/s/a/sakirilyuk/work/arch-pc/lab09/lab9-1.asm
#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]

```

Рис. 2.2: Ввод программы из листинга

```

sakirilyuk@dk5n60 ~/work/arch-pc/lab09 $ nasm -f elf lab9-1.asm
sakirilyuk@dk5n60 ~/work/arch-pc/lab09 $ ld -m elf_i386 -o lab9-1 lab9-1.o
sakirilyuk@dk5n60 ~/work/arch-pc/lab09 $ ./lab9-1
Введите N: 11
11
10
9
8
7
6
5
4
3
2
1

```

Рис. 2.3: Создание исполняемого файла

Меняем текст программы, добавив изменение значения регистра `ecx` в цикле (рис. 2.4), создаём исполняемый файл и проверяем его работу (рис. 2.5).

```

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

```

Рис. 2.4: Изменение текста файла

```

sakirilyuk@dk5n60 ~/work/arch-pc/lab09 $ nasm -f elf lab9-1.asm
sakirilyuk@dk5n60 ~/work/arch-pc/lab09 $ ld -m elf_i386 -o lab9-1 lab9-1.o
sakirilyuk@dk5n60 ~/work/arch-pc/lab09 $ ./lab9-1
Введите N: 

```

Рис. 2.5: Создание исполняемого файла

Данный пример показывает, что использование этого регистра в теле цикла может привести к некорректной работе программы. Число проходов не соответствует значению N, введённому с клавиатуры, программа заиклилась.

Снова вносим изменения в текст программы, добавив команды `push` и `pop` для сохранения значения счётчика цикла (рис. 2.6), и исполняем файл (рис. 2.7). В данном случае число проходов цикла равно значению N.



```

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

```

Рис. 2.6: Изменение текста файла

```

sakirilyuk@dk5n60 ~/work/arch-pc/lab09 $ nasm -f elf lab9-1.asm
sakirilyuk@dk5n60 ~/work/arch-pc/lab09 $ ld -m elf_i386 -o lab9-1 lab9-1.o
sakirilyuk@dk5n60 ~/work/arch-pc/lab09 $ ./lab9-1
Введите N: 11
10
9
8
7
6
5
4
3
2
1
0
Ошибка сегментирования (стек памяти сброшен на диск)
sakirilyuk@dk5n60 ~/work/arch-pc/lab09 $ 

```

Рис. 2.7: Исполнение файла

Создаём файл lab9-2.asm (рис. 2.8) и вводим в него текст программы из 2-го листинга(рис. 2.9). Создаём исполняемый файл и проверяем его работу (рис. 2.10), программой было обработано 4 аргумента.

```
sakirilyuk@dk5n60 ~ $ cd ~/work/arch-pc/lab09
sakirilyuk@dk5n60 ~/work/arch-pc/lab09 $ touch lab9-2.asm
sakirilyuk@dk5n60 ~/work/arch-pc/lab09 $
```

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 sprintLF
loop next

_end:
call quit
```

Рис. 2.9: Ввод текста файла

```
sakirilyuk@dk5n60 ~ $ cd ~/work/arch-pc/lab09
sakirilyuk@dk5n60 ~/work/arch-pc/lab09 $ touch lab9-2.asm
sakirilyuk@dk5n60 ~/work/arch-pc/lab09 $ mc

sakirilyuk@dk5n60 ~/work/arch-pc/lab09 $ nasm -f elf lab9-2.asm
sakirilyuk@dk5n60 ~/work/arch-pc/lab09 $ ./lab9-2 аргумент1 аргумент 2 'аргумент 3'
аргумент1
аргумент
2
аргумент 3
sakirilyuk@dk5n60 ~/work/arch-pc/lab09 $
```

Рис. 2.10: Создание исполняемого файла

Создаём файл lab8-3.asm (рис. 2.11), вводим текст программы из 3-го листинга (рис. 2.12), создаём исполняемый файл и проверяем его (рис. 2.13).

```
sakirilyuk@dk5n60 ~ $ cd ~/work/arch-pc/lab09
sakirilyuk@dk5n60 ~/work/arch-pc/lab09 $ touch lab9-3.asm
sakirilyuk@dk5n60 ~/work/arch-pc/lab09 $
```

Рис. 2.11: Создание нового файла

```

/afs/.dk.sci.pfu.edu.ru/home/s/a/sakirilyuk/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
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

```

Рис. 2.12: Ввод текста файла

```

sakirilyuk@dk5n60 ~/work/arch-pc/lab09 $ nasm -f elf lab9-3.asm
sakirilyuk@dk5n60 ~/work/arch-pc/lab09 $ ld -m elf_i386 -o lab9-3 lab9-3.o
sakirilyuk@dk5n60 ~/work/arch-pc/lab09 $ ./lab9-3 12 13 7 10 5
Результат: 47
sakirilyuk@dk5n60 ~/work/arch-pc/lab09 $ 

```

Рис. 2.13: Создание исполняемого файла

Изменяем текст программы из листинга №3 для вычисления произведения аргументов командной строки (рис. 2.14), создаём исполняемый файл и проверяем его работу (рис. 2.15).

```

%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

```

Рис. 2.14: Изменение текста файла

```

sakirilyuk@dk5n60 ~/work/arch-pc/lab09 $ nasm -f elf lab9-3.asm
sakirilyuk@dk5n60 ~/work/arch-pc/lab09 $ ld -m elf_i386 -o lab9-3 lab9-3.o
sakirilyuk@dk5n60 ~/work/arch-pc/lab09 $ ./lab9-3 12 2
Результат: 24
sakirilyuk@dk5n60 ~/work/arch-pc/lab09 $ 

```

Рис. 2.15: Создание исполняемого файла

### #Задание для самостоятельной работы

Изменяем текст программы из листинга №3 в соответствии с листингом №3 (рис. 2.16), вновь создаём исполняемый файл и проверяем его работу (рис. 2.17).

```

SECTION .text
global _start
_start:
    pop ecx

    pop edx

    sub ecx,1

    mov ebx,3

    mov esi,0

next:
    cmp ecx,0h
    jz _end

    pop eax
    call atoi
    add eax, 10

```

Рис. 2.16: Изменение текста файла

```

sakirilyuk@dk5n60 ~/work/arch-pc/lab09 $ nasm -f elf lab9-3.asm
sakirilyuk@dk5n60 ~/work/arch-pc/lab09 $ ld -m elf_i386 -o lab9-3 lab9-3.o
sakirilyuk@dk5n60 ~/work/arch-pc/lab09 $ ./lab9-3 1 2
Результат: 69
sakirilyuk@dk5n60 ~/work/arch-pc/lab09 $ 

```

Рис. 2.17: Создание исполняемого файла

## **3 Выводы**

В ходе работы я приобрела навыки написания программ с использованием циклов, обработкой аргументов командной строки.