

# **Отчет по лабораторной работе №10**

**дисциплина: Архитектура компьютера**

Белоусова Елизавета Валентиновна

# Содержание

## **Список иллюстраций**

# Список таблиц

## 1 Цель работы

Цель данной лабораторной работы - приобретение навыков написания программ для работы с файлами.

## 2 Задание

1. Выполнение лабораторной работы
2. Выполнение заданий для самостоятельной работы

## 3 Теоретическое введение

ОС GNU/Linux является многопользовательской операционной системой. И для обеспечения защиты данных одного пользователя от действий других пользователей существуют специальные механизмы разграничения доступа к файлам. Кроме ограничения доступа, данный механизм позволяет разрешить другим пользователям доступ данным для совместной работы. Права доступа определяют набор действий (чтение, запись, выполнение), разрешённых для выполнения пользователям системы над файлами. Для каждого файла пользователь может входить в одну из трех групп: владелец, член группы владельца, все остальные. Для каждой из этих групп может быть установлен свой набор прав доступа. Владелец файла является его создатель. Для предоставления прав доступа другому пользователю или другой группе командой `chown [ключи] [:новая_группа]` или `chgrp [ключи] < новая_группа >` В операционной системе Linux существуют различные методы управления файлами, например, такие как создание и открытие файла, только для чтения или для чтения и записи, добавления в существующий файл, закрытия и удаления файла, предоставление прав доступа. Обработка файлов в операционной системе Linux осуществляется за счет использования

определенных системных вызовов. Для корректной работы и доступа к файлу при его открытии или создании, файлу присваивается уникальный номер (16-битное целое число) – дескриптор файла. Для создания и открытия файла служит системный вызов `sys_creat`, который использует следующие аргументы: права доступа к файлу в регистре `ECX`, имя файла в `EBX` и номер системного вызова `sys_creat` (8) в `EAX`. Для открытия существующего файла служит системный вызов `sys_open`, который использует следующие аргументы: права доступа к файлу в регистре `EDX`, режим доступа к файлу в регистр `ECX`, имя файла в `EBX` и номер системного вызова `sys_open` (5) в `EAX`. Для записи в файл служит системный вызов `sys_write`, который использует следующие аргументы: количество байтов для записи в регистре `EDX`, строку содержимого для записи `ECX`, файловый дескриптор в `EBX` и номер системного вызова `sys_write` (4) в `EAX`. Системный вызов возвращает фактическое количество записанных байтов в регистр `EAX`. В случае ошибки, код ошибки также будет находиться в регистре `EAX`. Прежде чем записывать в файл, его необходимо создать или открыть, что позволит получить дескриптор файла. Для чтения данных из файла служит системный вызов `sys_read`, который использует следующие аргументы: количество байтов для чтения в регистре `EDX`, адрес в памяти для записи прочитанных данных в `ECX`, файловый дескриптор в `EBX` и номер системного вызова `sys_read` (3) в `EAX`. Как и для записи, прежде чем читать из файла, его необходимо открыть, что позволит получить дескриптор файла. Для правильного закрытия файла служит системный вызов `sys_close`, который использует один аргумент – дескриптор файла в регистре `EBX`. После вызова ядра происходит удаление дескриптора файла, а в случае ошибки, системный вызов возвращает код ошибки в регистр `EAX`. Для изменения содержимого файла служит системный вызов `sys_lseek`, который использует следующие аргументы: исходная позиция для смещения `EDX`, значение смещения в байтах в `ECX`, файловый дескриптор в `EBX` и номер системного вызова `sys_lseek` (19) в `EAX`. Удаление файла осуществляется системным вызовом `sys_unlink`, который использует один аргумент – имя файла в регистре `EBX`.

## 4 Выполнение лабораторной работы

### 1) Выполнение лабораторной работы

Перехожу в каталог, созданный для файлов с программами для лабораторной работы №10 (рис.1)

```
evbelousova@dk4n69 ~ $ cd ~/work/study/2023-2024/"Архитектура компьютера"/arch-pc/labs/lab10
evbelousova@dk4n69 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab10 $
```

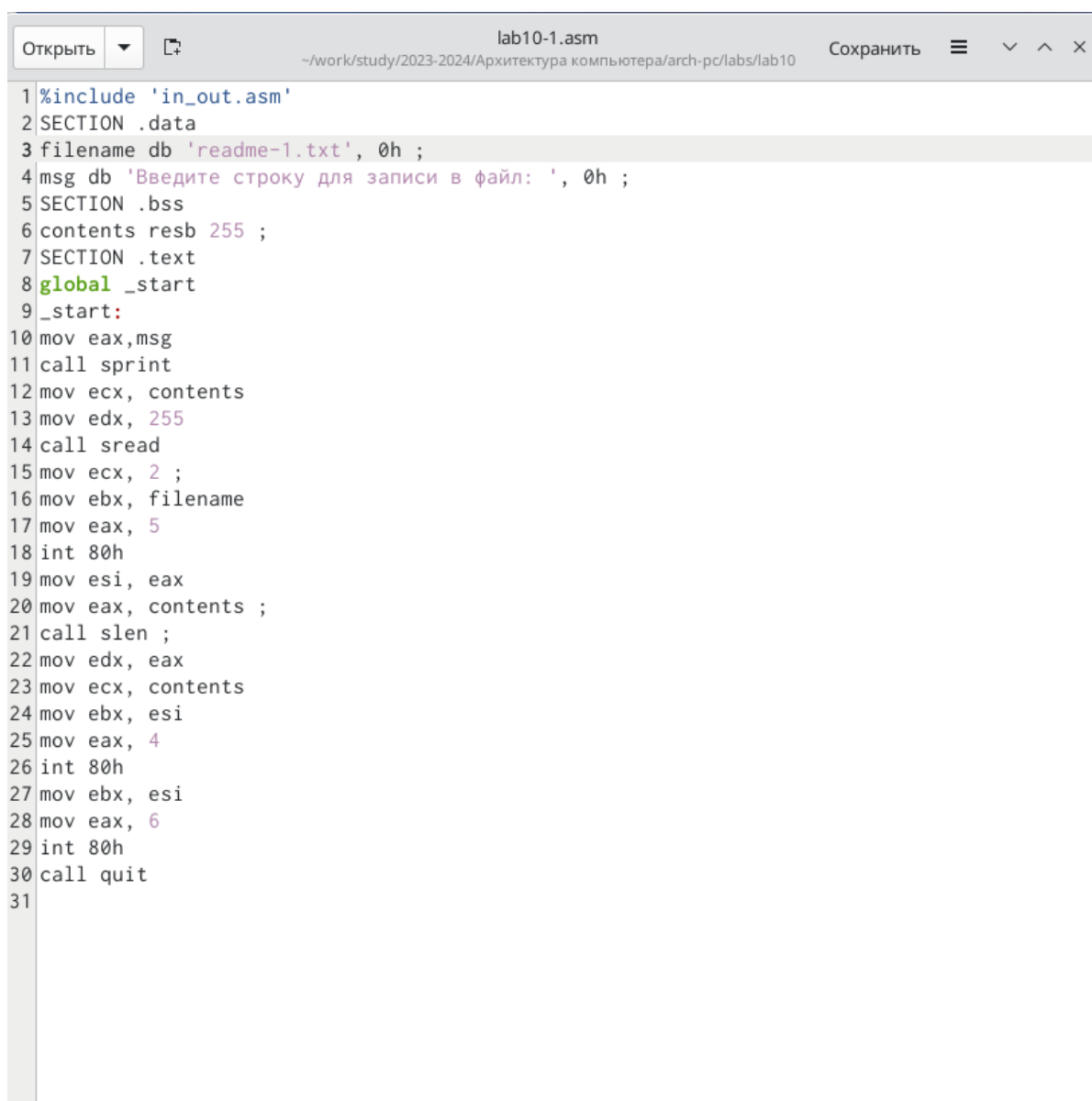
Рис. 1: Перемещение между директориями

С помощью команды touch создаю файлы lab10-1.asm, readme-1.txt и readme-2.txt. Копирую в текущий каталог файл in\_out.asm, так как он будет использоваться в дальнейшем (рис. 2).

```
evbelousova@dk4n69 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab10 $ touch lab10-1.asm readme-1.txt readme-2.txt
```

Рис. 1: Создание файлов

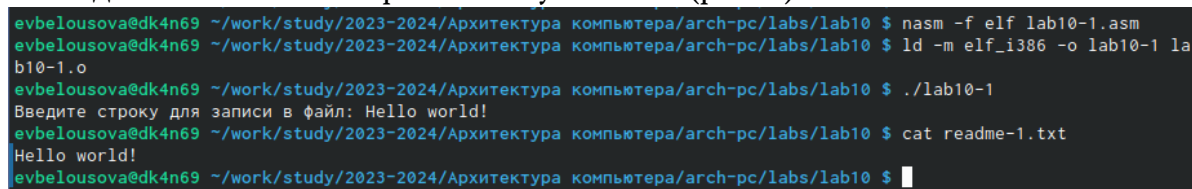
Открываю созданный файл lab10-1.asm, вставляю в него программу из листинга 10.1 (рис. 3)



```
1 %include 'in_out.asm'
2 SECTION .data
3 filename db 'readme-1.txt', 0h ;
4 msg db 'Введите строку для записи в файл: ', 0h ;
5 SECTION .bss
6 contents resb 255 ;
7 SECTION .text
8 global _start
9 _start:
10 mov eax, msg
11 call sprint
12 mov ecx, contents
13 mov edx, 255
14 call sread
15 mov ecx, 2 ;
16 mov ebx, filename
17 mov eax, 5
18 int 80h
19 mov esi, eax
20 mov eax, contents ;
21 call slen ;
22 mov edx, eax
23 mov ecx, contents
24 mov ebx, esi
25 mov eax, 4
26 int 80h
27 mov ebx, esi
28 mov eax, 6
29 int 80h
30 call quit
31
```

Рис. 3: Редактирование файла

Создаю исполняемый файл и запускаю его (рис. 3).



```
evbelousova@dk4n69 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab10 $ nasm -f elf lab10-1.asm
evbelousova@dk4n69 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab10 $ ld -m elf_i386 -o lab10-1 la
b10-1.o
evbelousova@dk4n69 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab10 $ ./lab10-1
Введите строку для записи в файл: Hello world!
evbelousova@dk4n69 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab10 $ cat readme-1.txt
Hello world!
evbelousova@dk4n69 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab10 $
```

Рис. 4: Запуск исполняемого файла

С помощью команды `chmod` изменяю права доступа к исполняемому файлу `lab10-1`, запрещая его выполнение (рис. 5).

```

evbelousova@dk4n69 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab10 $ chmod u-x lab10-1
evbelousova@dk4n69 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab10 $ ./lab10-1
bash: ./lab10-1: Отказано в доступе
evbelousova@dk4n69 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab10 $

```

Рис. 5: Изменение прав доступа к файлу

Файл не выполняется, так как я запретила доступ к нему, указав в команде “-x”.

С помощью команды `chmod` изменяю права доступа к файлу `lab10-1.asm` с исходным текстом программы, добавив права на исполнение (рис. 6).

```

evbelousova@dk4n69 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab10 $ chmod u+x lab10-1.asm
evbelousova@dk4n69 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab10 $ ./lab10-1.asm
./lab10-1.asm: строка 1: fg: нет управления заданиями
./lab10-1.asm: строка 2: SECTION: команда не найдена
./lab10-1.asm: строка 3: filename: команда не найдена
./lab10-1.asm: строка 4: msg: команда не найдена
./lab10-1.asm: строка 5: SECTION: команда не найдена
./lab10-1.asm: строка 6: contents: команда не найдена
./lab10-1.asm: строка 7: SECTION: команда не найдена
./lab10-1.asm: строка 8: global: команда не найдена
./lab10-1.asm: строка 9: _start:: команда не найдена
./lab10-1.asm: строка 10: mov: команда не найдена
./lab10-1.asm: строка 11: call: команда не найдена
./lab10-1.asm: строка 12: mov: команда не найдена
./lab10-1.asm: строка 13: mov: команда не найдена
./lab10-1.asm: строка 14: call: команда не найдена
./lab10-1.asm: строка 15: mov: команда не найдена
./lab10-1.asm: строка 16: mov: команда не найдена
./lab10-1.asm: строка 17: mov: команда не найдена
./lab10-1.asm: строка 18: int: команда не найдена
./lab10-1.asm: строка 19: mov: команда не найдена
./lab10-1.asm: строка 20: mov: команда не найдена
./lab10-1.asm: строка 21: call: команда не найдена
./lab10-1.asm: строка 22: mov: команда не найдена
./lab10-1.asm: строка 23: mov: команда не найдена
./lab10-1.asm: строка 24: mov: команда не найдена
./lab10-1.asm: строка 25: mov: команда не найдена

```

Рис. 6: Изменение прав доступа к файлу

Файл начинает исполняться, однако он не содержит в себе команд для терминала, поэтому не может быть исполнен.

В соответствии со своим вариантом (18) предоставляю права доступа к файлу `readme-1.txt` представленные в символьном виде, а для файла `readme-2.txt` – в двоичном виде. Проверяю правильность выполнения с помощью команды `ls -l` (рис.7)



```

evbelousova@dk4n69 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab10 $ chmod 640 readme-1.txt # -wx
r-x -wx
evbelousova@dk4n69 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab10 $ chmod 640 readme-2.txt # 101
011 110
evbelousova@dk4n69 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab10 $ ls -l
итого 35
-rw-r--r-- 1 evbelousova studsci 3942 ноя 24 18:23 in_out.asm
-rwxr-xr-x 1 evbelousova studsci 9164 дек 12 13:41 lab10-1
-rwxr--r-- 1 evbelousova studsci 487 дек 12 13:40 lab10-1.asm
-rw-r--r-- 1 evbelousova studsci 13468 дек 12 13:16 lab10-1.lst
-rw-r--r-- 1 evbelousova studsci 1472 дек 12 13:41 lab10-1.o
drwxr-xr-x 3 evbelousova studsci 2048 сен 29 19:21 presentation
-rw-r----- 1 evbelousova studsci 13 дек 12 13:51 readme-1.txt
-rw-r----- 1 evbelousova studsci 0 дек 12 13:13 readme-2.txt
drwxr-xr-x 5 evbelousova studsci 2048 сен 29 19:21 report
evbelousova@dk4n69 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab10 $

```

Рис. 7: Предоставление прав доступа к файлам

## 2) Выполнение заданий для самостоятельной работы

Создаю файлы lab10-2.asm и name.txt (рис. 8)

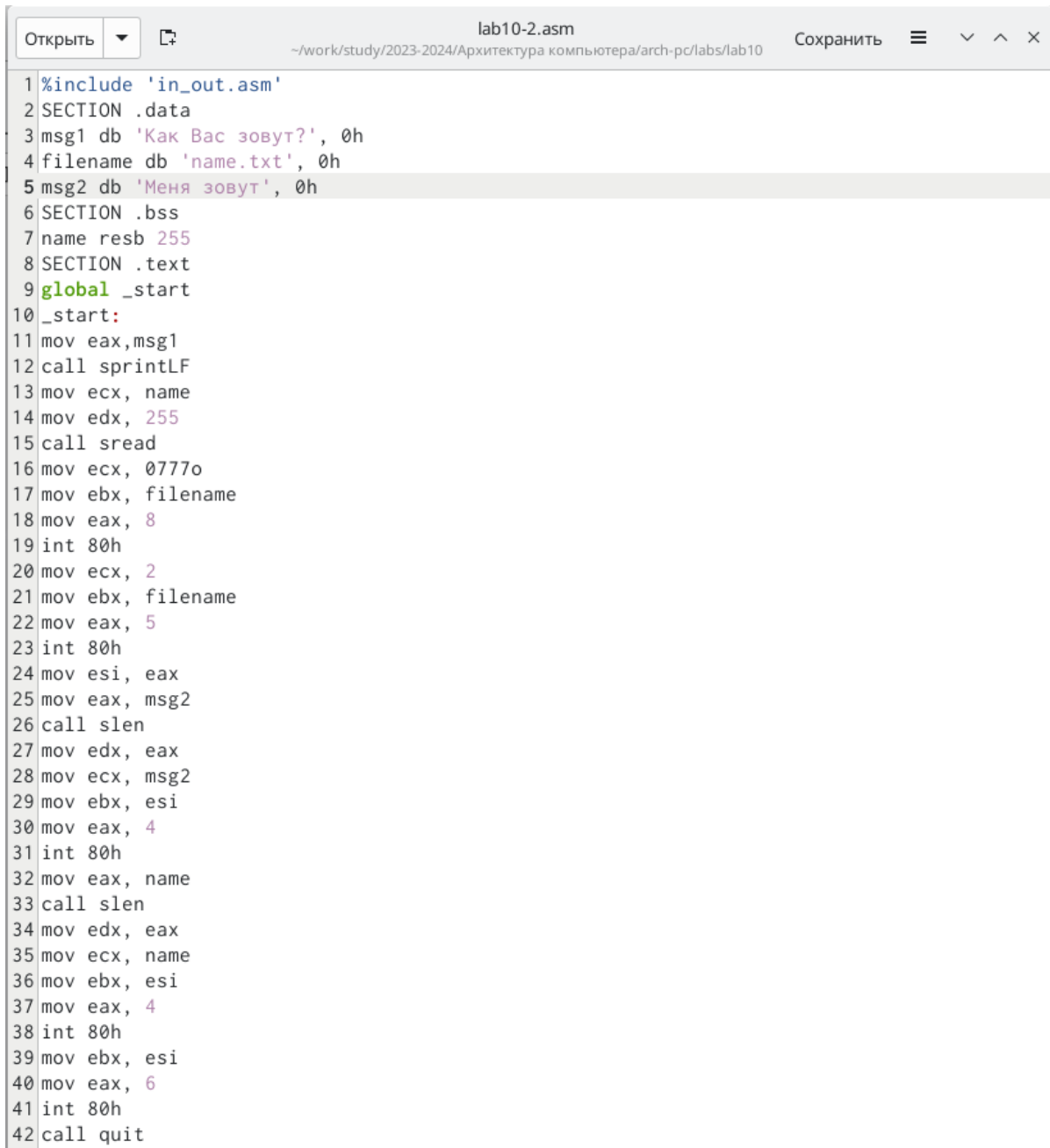
```

evbelousova@dk4n69 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab10 $ touch lab10-2.asm
evbelousova@dk4n69 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab10 $ touch name.txt
evbelousova@dk4n69 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab10 $

```

Рис. 8: Создание файлов

Открываю созданный файл и пишу в нем программу, позволяющую вывести на экран вопрос “Как Вас зовут”, ввести с клавиатуры свое имя, а затем создающую файл, который содержит сообщение “Меня зовут ФИ” (рис. 9).



```
1 %include 'in_out.asm'
2 SECTION .data
3 msg1 db 'Как Вас зовут?', 0h
4 filename db 'name.txt', 0h
5 msg2 db 'Меня зовут', 0h
6 SECTION .bss
7 name resb 255
8 SECTION .text
9 global _start
10 _start:
11 mov eax, msg1
12 call sprintLF
13 mov ecx, name
14 mov edx, 255
15 call sread
16 mov ecx, 0777o
17 mov ebx, filename
18 mov eax, 8
19 int 80h
20 mov ecx, 2
21 mov ebx, filename
22 mov eax, 5
23 int 80h
24 mov esi, eax
25 mov eax, msg2
26 call slen
27 mov edx, eax
28 mov ecx, msg2
29 mov ebx, esi
30 mov eax, 4
31 int 80h
32 mov eax, name
33 call slen
34 mov edx, eax
35 mov ecx, name
36 mov ebx, esi
37 mov eax, 4
38 int 80h
39 mov ebx, esi
40 mov eax, 6
41 int 80h
42 call quit
```

Рис. 9: Редактирование файла

Создаю исполняемый файл и запускаю его. Вижу, что программа работает корректно. (рис. 10)

```

evbelousova@dk4n69 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab10 $ nasm -f elf lab10-2.asm
evbelousova@dk4n69 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab10 $ ld -m elf_i386 -o lab10-2 la
b10-2.o
evbelousova@dk4n69 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab10 $ ls
in_out.asm lab10-1.asm lab10-1.o lab10-2.asm name.txt readme-1.txt report
lab10-1 lab10-1.lst lab10-2 lab10-2.o presentation readme-2.txt
evbelousova@dk4n69 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab10 $ ./lab10-2
Как Вас зовут?
Белоусова Елизавета
evbelousova@dk4n69 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab10 $ cat name.txt
Меня зовут Белоусова Елизавета
evbelousova@dk4n69 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab10 $

```

Рис. 10: Запуск программы

Листинг 10.2 (написанная программа)

```

%include 'in_out.asm'

SECTION .data
msg1 db 'Как Вас зовут?', 0h
filename db 'name.txt', 0h
msg2 db 'Меня зовут', 0h

SECTION .bss
name resb 255

SECTION .text
global _start
_start:
mov eax, msg1
call sprintLF
mov ecx, name
mov edx, 255
call sread
mov ecx, 0777o
mov ebx, filename
mov eax, 8
int 80h
mov ecx, 2
mov ebx, filename
mov eax, 5

```

```
int 80h
mov esi, eax
mov eax, msg2
call slen
mov edx, eax
mov ecx, msg2
mov ebx, esi
mov eax, 4
int 80h
mov eax, name
call slen
mov edx, eax
mov ecx, name
mov ebx, esi
mov eax, 4
int 80h
mov ebx, esi
mov eax, 6
int 80h
call quit
```

## **5 Выводы**

При выполнении данной лабораторной работы, я приобрела практические навыки написания программ для работы с файлами.

## **6 Список литературы**

Архитектура ЭВМ