

Отчет по лабораторной работе №5

Дисциплина: архитектура компьютера

Ермакова Анастасия Алексеевна

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	8
4.1	Основы работы с mc	8
4.2	Структура программы на языке ассемблера NASM	9
4.3	Подключение внешнего файла	12
4.4	Выполнение заданий для самостоятельной работы	14
5	Выводы	18
6	Список литературы	19

Список иллюстраций

4.1	Команда <code>mc</code>	8
4.2	Интерфейс Midnight Commander	8
4.3	Создание папки в каталоге	9
4.4	Создание файла с помощью команды <code>touch</code>	9
4.5	Окно Midnight Commander. Редактор <code>mcedit</code>	10
4.6	Файл <code>lab5-1.asm</code>	10
4.7	Создание объектного файла	11
4.8	Компановка объектного файла в исполняемый	11
4.9	Запуск исполняемого файла	11
4.10	Работа программы исполняемого файла	11
4.11	Файл <code>in_out.asm</code> в каталоге Загрузки	12
4.12	Копирование файла в нужный каталог	12
4.13	Файл <code>in_out.asm</code> в каталоге <code>lab05</code>	13
4.14	Изменение текста файла <code>lab5-2.asm</code>	13
4.15	Работа программы исполняемого файла	14
4.16	Работа измененной программы исполняемого файла	14
4.17	Копия файла <code>lab5-1.asm</code>	14
4.18	Текст программы файла <code>lab5-1-1.asm</code>	15
4.19	Работа программы исполняемого файла <code>lab5-1-1</code>	15
4.20	Работа программы исполняемого файла <code>lab5-2-1</code>	17

Список таблиц

1 Цель работы

Цель данной лабораторной работы - приобрести практические навыки работы в Midnight Commander, освоить инструкции языка ассемблера mov и int.

2 Задание

1. Основы работы с тс
2. Структура программы на языке ассемблера NASM
3. Подключение внешнего файла
4. Выполнение заданий для самостоятельной работы

3 Теоретическое введение

Midnight Commander (или просто mc) — это программа, которая позволяет просматривать структуру каталогов и выполнять основные операции по управлению файловой системой, т.е. mc является файловым менеджером. Midnight Commander позволяет сделать работу с файлами более удобной и наглядной. Для активации оболочки Midnight Commander достаточно ввести в командной строке mc и нажать клавишу Enter. В Midnight Commander используются функциональные клавиши F1 — F10, к которым привязаны часто выполняемые операции.

4 Выполнение лабораторной работы

4.1 Основы работы с mc

Открываю Midnight Commander. Для этого прописываю в терминале mc. (рис. 4.1).

```
anastasia@fedora:~$ mc
```

Рис. 4.1: Команда mc

Пользуясь курсорными клавишами и Enter перехожу в каталог ~/work/arch-pc. (рис. 4.2).

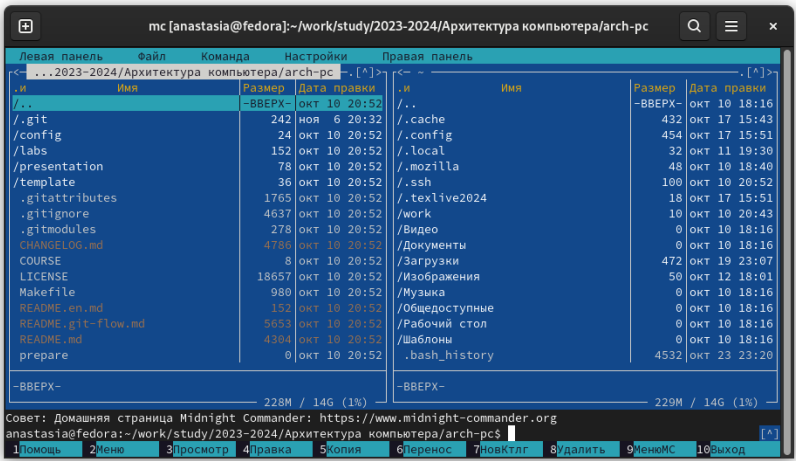


Рис. 4.2: Интерфейс Midnight Commander

С помощью функциональной клавиши F7 создаю папку lab05 и перехожу в созданный каталог. (рис. 4.3).

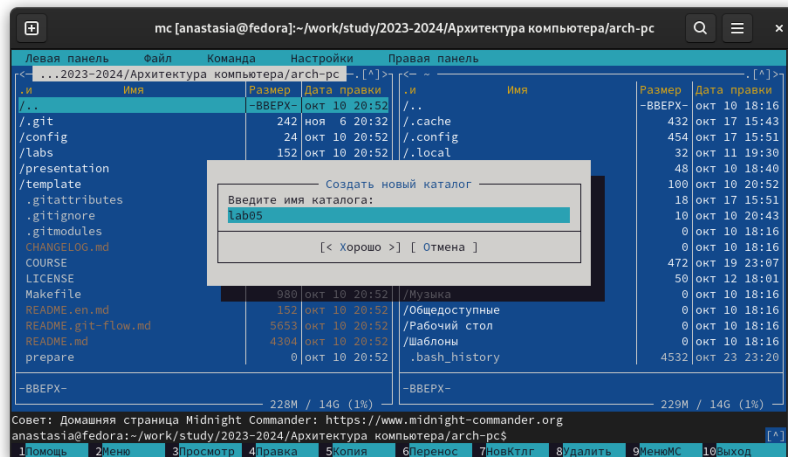


Рис. 4.3: Создание папки в каталоге

Пользуясь строкой ввода и командой `touch` создаю файл `lab5-1.asm`. (рис. 4.4).

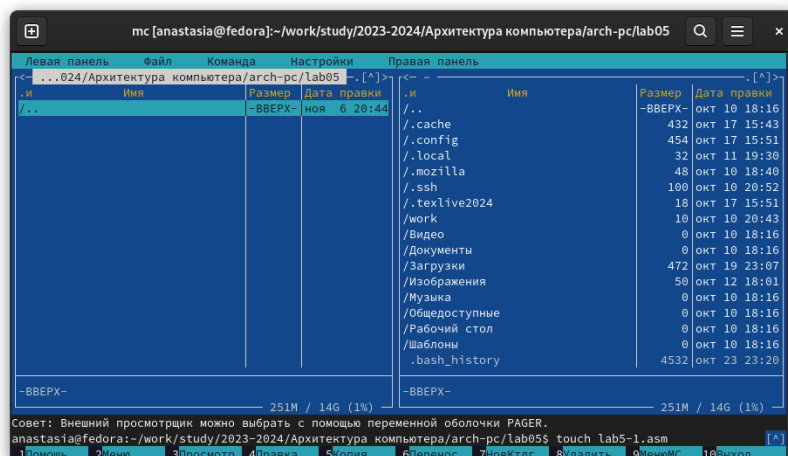


Рис. 4.4: Создание файла с помощью команды `touch`

4.2 Структура программы на языке ассемблера NASM

С помощью функциональной клавиши `F4` открываю файл `lab5-1.asm` для редактирования во встроенном редакторе `mcedit`. Ввожу текст программы из листинга 5.1, сохраняю изменения и закрываю файл. (рис. 4.5).

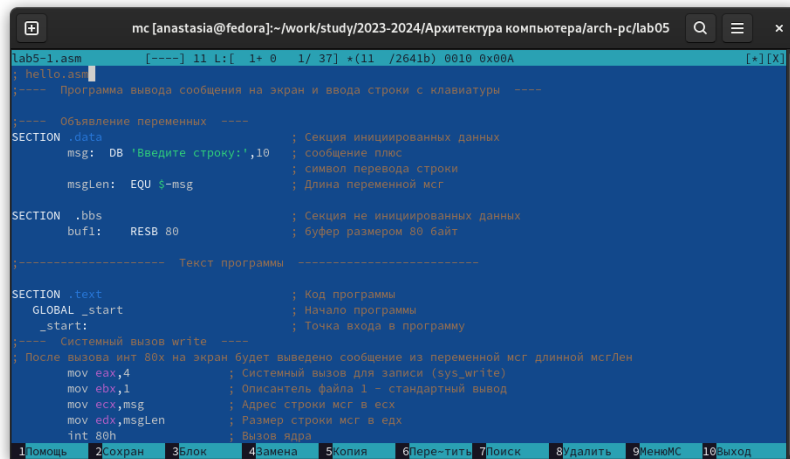


Рис. 4.5: Окно Midnight Commander. Редактор mcedit

С помощью функциональной клавиши F3 открываю файл lab5-1.asm для просмотра и убеждаюсь, что файл содержит текст программы. (рис. 4.6).

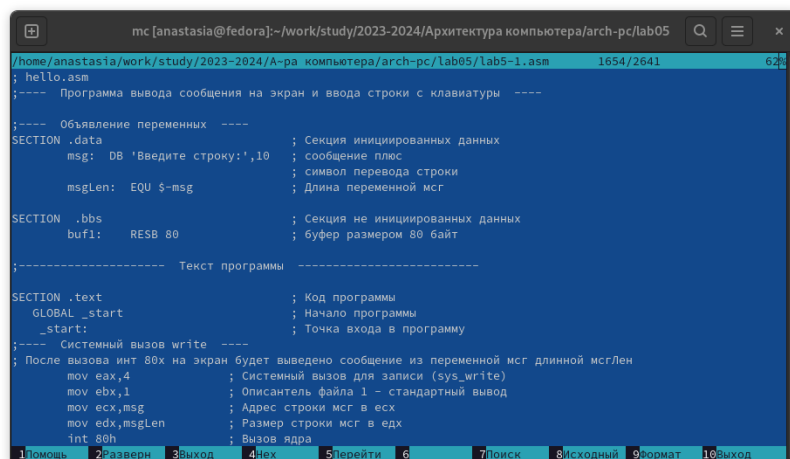


Рис. 4.6: Файл lab5-1.asm

Оттранслирую текст программы lab5-1.asm в объектный файл, для этого прописываю `nasm -f elf lab5-1.asm` (рис. 4.7). Затем выполняю компоновку объектного файла (`ld -m elf_i386 -o lab5-1 lab5-1.o`) (рис. 4.8), созданся исполняемый файл `*lab5-1`. Запускаю получившийся файл. (рис. 4.9).

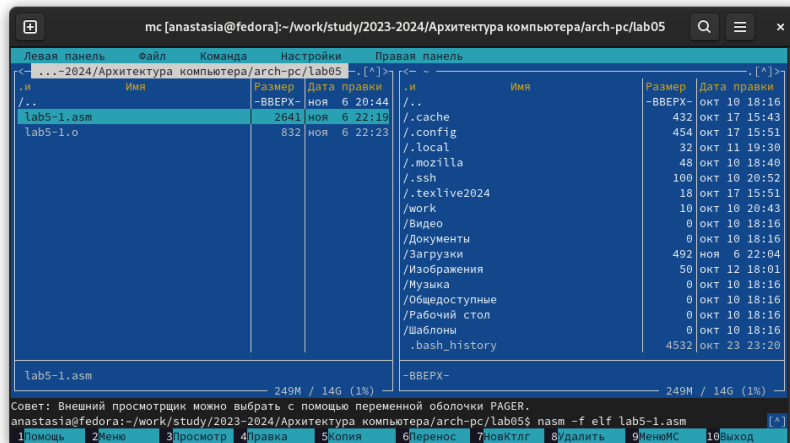


Рис. 4.7: Создание объектного файла

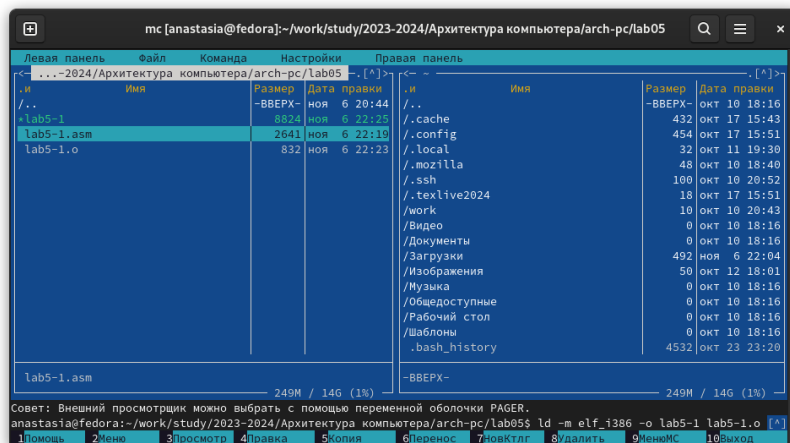


Рис. 4.8: Компановка объектного файла в исполняемый

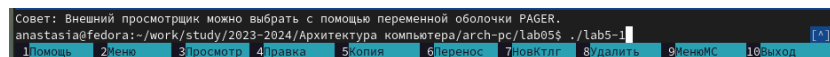


Рис. 4.9: Запуск исполняемого файла

Программа вывела строку 'Введите строку:', ввожу свои ФИО. На этом работа программы заканчивается. (рис. 4.10).

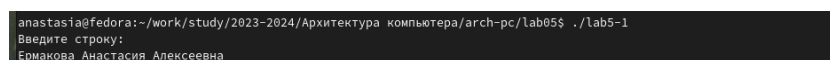


Рис. 4.10: Работа программы исполняемого файла

4.3 Подключение внешнего файла

Скачиваю файл in_out.asm со страницы курса в ТУИС. Он автоматически сохраняется в каталог Загрузки. (рис. 4.11).

Левая панель				Файл	Команда	Настройки	
< ~/Загрузки							. [^]>
				Имя	Размер	Дата	правки
/..				-ВВЕРХ-		ноя 7 13:46	
/install-tl-20241010				132		окт 10 03:34	
/фото				44		ноя 6 22:31	
Attachments_kno~19_20-53-56.zip				746084		окт 19 21:03	
in_out.asm				3942		ноя 6 22:04	
install-tl-unx.tar.gz				5756463		окт 10 21:09	
pandoc-3.5-linux-amd64.tar.gz				31612K		окт 17 15:25	
pandoc-crossref~x-X64.tar(1).xz				7337164		окт 17 15:42	

Рис. 4.11: Файл in_out.asm в каталоге Загрузки

Копирую его с помощью функциональной клавиши F5 в каталог ~/work/arch-
pc/lab05 в котором работаю, и в котором лежит файл lab5-1.asm. (рис. 4.12 - 4.13).

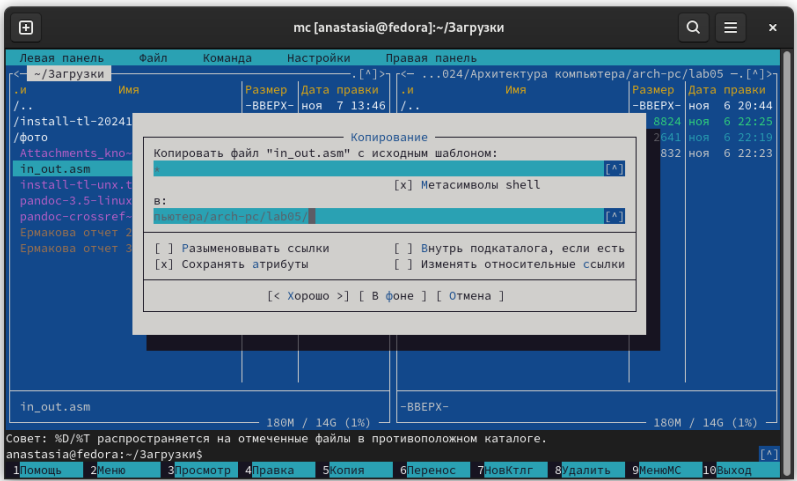


Рис. 4.12: Копирование файла в нужный каталог

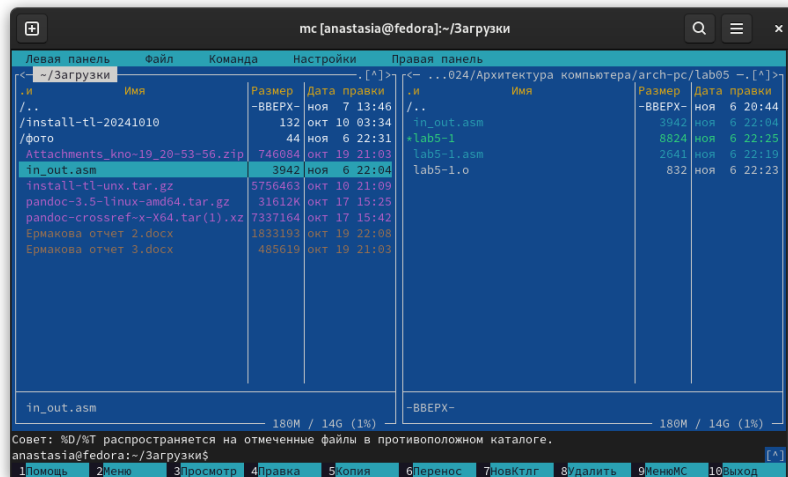


Рис. 4.13: Файл in_out.asm в каталоге lab05

С помощью функциональной клавиши F6 создаю копию файла lab5-1.asm с именем lab5-2.asm. Исправляю текст программы в последнем файле с использованием программ из внешнего файла in_out.asm в соответствии с листингом 5.2. (рис. 4.14).

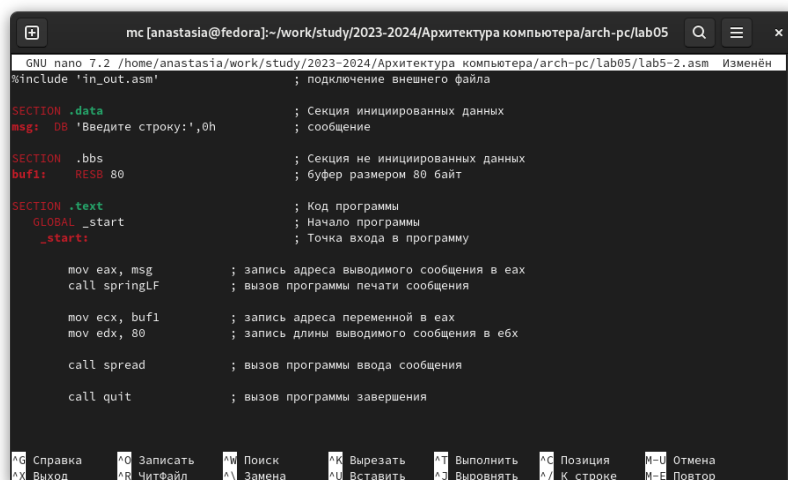


Рис. 4.14: Изменение текста файла lab5-2.asm

Создаю исполняемый файл и проверяю его работу. (рис. 4.15).

```
anastasia@fedora:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab05$ ./lab5-2
Введите строку:
Ермакова Анастасия Алексеевна
```

Рис. 4.15: Работа программы исполняемого файла

Заменяю в файле lab5-2.asm подпрограмму sprintLF на sprint. Создаю исполняемый файл и проверяю его работу. (рис. 4.16).

```
anastasia@fedora:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab05$ ./lab5-2
Введите строку: Ермакова Анастасия Алексеевна
```

Рис. 4.16: Работа измененной программы исполняемого файла

Разница между ними в том, что первая программа запрашивает ввод с новой строки (sptintLF), а вторая запрашивает ввод без перехода на новую строку(sprint).

4.4 Выполнение заданий для самостоятельной работы

Создаю копию файла lab5-1.asm с именем lab5-1-1.asm. (рис. 4.17).

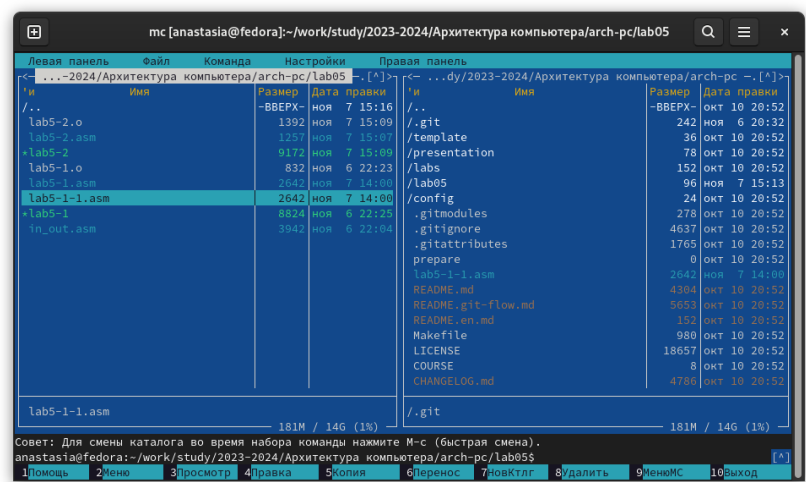
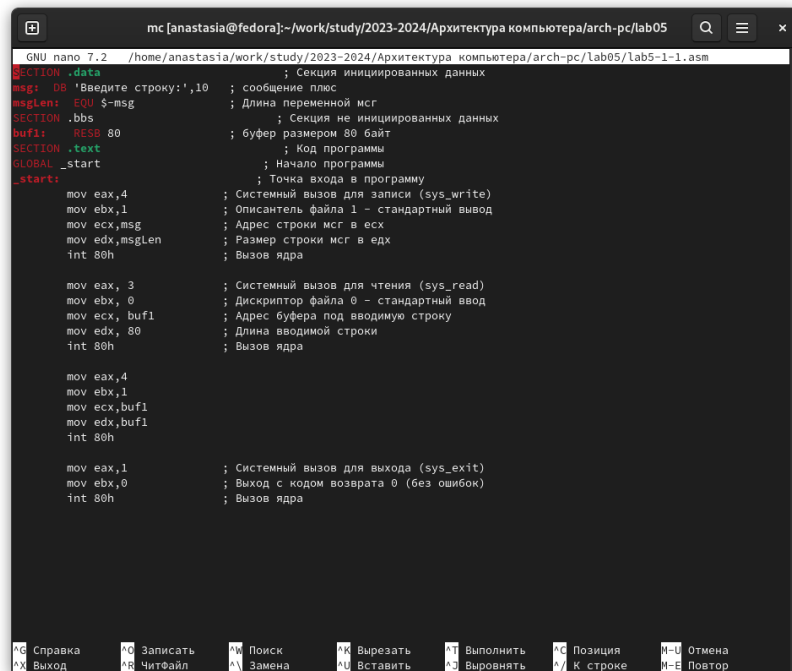


Рис. 4.17: Копия файла lab5-1.asm

С помощью функциональной клавиши F4 открываю созданный файл для редактирования. Изменяю программу (без использования внешнего файла) так,

чтобы после того, как я введу строку с клавиатуры, она выводила ее на экран.
(рис. 4.18).



```
GNU nano 7.2 /home/anastasia/work/study/2023-2024/Архитектура компьютера/arch-pc/lab05/lab5-1-1.asm
SECTION .data                                ; Секция иницированных данных
msg: DB 'Введите строку:',10                ; сообщение плюс
msgLen: EQU $-msg                           ; Длина переменной msg
SECTION .bss                                ; Секция не иницированных данных
buf1: RESB 80                               ; буфер размером 80 байт
SECTION .text                               ; Код программы
GLOBAL _start                               ; Начало программы
_start:                                     ; Точка входа в программу
    mov eax,4                               ; Системный вызов для записи (sys_write)
    mov ebx,1                               ; Файл 1 - стандартный вывод
    mov ecx,msg                             ; Адрес строки msg в ecx
    mov edx,msgLen                          ; Размер строки msg в edx
    int 80h                                 ; Вызов ядра

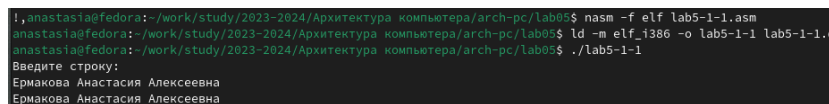
    mov eax,3                               ; Системный вызов для чтения (sys_read)
    mov ebx,0                               ; Файл 0 - стандартный ввод
    mov ecx,buf1                            ; Адрес буфера под вводимую строку
    mov edx,80                              ; Длина вводимой строки
    int 80h                                 ; Вызов ядра

    mov eax,4                               ; Системный вызов для записи (sys_write)
    mov ebx,1                               ; Файл 1 - стандартный вывод
    mov ecx,buf1                            ; Адрес буфера под вводимую строку
    mov edx,buf1                            ; Длина вводимой строки
    int 80h                                 ; Вызов ядра

    mov eax,1                               ; Системный вызов для выхода (sys_exit)
    mov ebx,0                               ; Выход с кодом возврата 0 (без ошибок)
    int 80h                                 ; Вызов ядра
```

Рис. 4.18: Текст программы файла lab5-1-1.asm

Получаю исполняемый файл и проверяю его работу: на приглашение ввести строку ввожу свои ФИО. (рис. 4.19).



```
I,anastasia@fedora: ~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab05$ nasm -f elf lab5-1-1.asm
anastasia@fedora: ~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab05$ ld -m elf_i386 -o lab5-1-1 lab5-1-1.o
anastasia@fedora: ~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab05$ ./lab5-1-1
Введите строку:
Ермакова Анастасия Алексеевна
Ермакова Анастасия Алексеевна
```

Рис. 4.19: Работа программы исполняемого файла lab5-1-1

Код программы:

```
SECTION .data                                ; Секция иницированных данных
msg: DB 'Введите строку:',10                ; сообщение плюс
msgLen: EQU $-msg                           ; Длина переменной msg
SECTION .bss                                ; Секция не иницированных данных
buf1: RESB 80                               ; буфер размером 80 байт
```

```

SECTION .text                ; Код программы
GLOBAL _start                ; Начало программы
_start:                      ; Точка входа в программу

mov eax,4                    ; Системный вызов для записи (sys_write)
mov ebx,1                    ; Описатель файла 1 - стандартный вывод
mov ecx,msg                  ; Адрес строки мсг в ecx
mov edx,msgLen               ; Размер строки мсг в edx
int 80h                      ; Вызов ядра

mov eax, 3                   ; Системный вызов для чтения (sys_read)
mov ebx, 0                   ; Дискриптор файла 0 - стандартный ввод
mov ecx, buf1                ; Адрес буфера под вводимую строку
mov edx, 80                  ; Длина вводимой строки
int 80h                      ; Вызов ядра

mov eax,4                    ; Системный вызов для записи
mov ebx,1                    ; Описатель файла 1 - стандартный вывод
mov ecx,buf1                 ; Адрес строки буф1 в ecx
mov edx,buf1                 ; Размер строки буф1
int 80h                      ; Вызов ядра

mov eax,1                    ; Системный вызов для выхода (sys_exit)
mov ebx,0                    ; Выход с кодом возврата 0 (без ошибок)
int 80h                      ; Вызов ядра

```

Создаю копию файла lab5-2.asm с именем lab5-2-1.asm. Изменяю эту программу (с использованием подпрограмм внешнего файла in_out.asm) так, чтобы она работала таким же образом, как и lab5-1-1. Создаю исполняемый файл и проверяю его работу. (рис. 4.20).


```

anastasia@fedora: ~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab0$ nasm -f elf lab5-2-1.asm
anastasia@fedora: ~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab0$ ld -m elf_i386 -o lab5-2-1 lab5-2-1.o
anastasia@fedora: ~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab0$ ./lab5-2-1
Введите строку: Ермакова Анастасия Алексеевна
Ермакова Анастасия Алексеевна
anastasia@fedora: ~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab0$

```

Рис. 4.20: Работа программы исполняемого файла lab5-2-1

Код программы:

```

#include 'in_out.asm'           ; подключение внешнего файла

SECTION .data                  ; Секция инициированных данных
msg:  DB 'Введите строку:',0h   ; сообщение
SECTION .bss                   ; Секция не инициированных данных
buf1:  RESB 80                 ; буфер размером 80 байт
SECTION .text                  ; Код программы
    GLOBAL _start              ; Начало программы
    _start:                   ; Точка входа в программу

mov eax, msg                   ; запись адреса выводимого сообщения в eax
call sprint                    ; вызов программы печати сообщения

mov ecx, buf1                  ; запись адреса переменной в eax
mov edx, 80                    ; запись длины выводимого сообщения в ebx

call sread                     ; вызов программы ввода сообщения

mov eax,4                      ; системный вызов для записи
mov ebx,1                      ; дескриптор файла 1 - стандартный вывод
mov ecx,buf1                   ; адрес строки buf1 в ecx
int 80h                        ; вызов ядра

call quit                      ; вызов подпрограммы завершения

```

5 Выводы

В ходе выполнения данной лабораторной работы я приобрела практические навыки работы в Midnight Commander и освоила инструкции языка ассемблера `mov` и `int`.

6 Список литературы

1. Архитектура ЭВМ