

Отчёт по лабораторной работе №5

Дисциплина: архитектура компьютера

Худдыева Дженнет

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	9
4.1	Структура программы на языке ассемблера NASM	11
4.2	Подключение внешнего файла	13
4.3	Выполнение заданий для самостоятельной работы	16
5	Выводы	21

Список иллюстраций

4.1	Открытый тс	9
4.2	Перемещение между директориями	10
4.3	Создание каталога	10
4.4	Перемещение между директориями	11
4.5	Редактирование файла	12
4.6	Открытие файла для просмотра	12
4.7	Копирование файла	13
4.8	Копирование файла	14
4.9	Редактирование файла	14
4.10	Исполнение файла	15
4.11	Отредактированный файл	15
4.12	Исполнение файла	15
4.13	Копирование файла	16
4.14	Редактирование файла	17
4.15	Исполнение файла	17
4.16	Копирование файла	19
4.17	Редактирование файла	19
4.18	Исполнение файла	20

Список таблиц

1 Цель работы

Целью данной лабораторной работы является приобретение практических навыков работы в Midnight Commander, освоение инструкций 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 файловом менеджером.Midnight Commander позволяет сделать работу с файлами более удобной и наглядной.Программа на языке ассемблера Nasm,как правило,состоит из трёх секций:секция кода программы (SECTION .txt) секция инициированных (известных во время компиляции) данных (SECTION .data) и секция не инициализированных данных (тех, под которые во время компиляции только отводится память,а значение присваивается в ходе выполнения программы) (SECTION .bss). Для объявления инициированных данных в секции .data используются директивы DB,DD,DQ,DT,которые резервируют память и указывают, какие значение должны хранится в этой памяти: -DB (define byte)-определяет переменную размером в 1 байт -DW (define word)-определяет переменную размером в 2 байта (слово) -DD (define double word)-определяет переменную размером в 4 байта (двойное слово) -DQ (define quad word)-определяет переменную размером в 8 байта(учетверённое слово) -DT (define ten bytes)-определяет переменную размером в 10 байта.Директивы используются для объявления простых переменных и для объявления массивов.Для определения строк принято использовать директиву DB в связи с особенностями хранения данных в оперативной памяти.Инструкция языка ассемблера mov предназначена для дублирования данных источника в приёмнике.

```
mov dst,src
```

Здесь операнд dst-приёмник, а src-источник. В качестве операнда могут вы-

ступать регистры(register),ячейки памяти (memory) и непосредственные значения(const). Инструкция языка ассемблера `int` предназначена для вызова прерывания с указанным номером.

`int n`

Здесь n-номер прерывания,принадлежащий диапазону 0-255.При программировании в Linux с использованием ядра `sys_calls` `n=80h` (принято задавать в шестнадцатеричной системе счисления)

4 Выполнение лабораторной работы

Открываю Midnight Commander, введя в терминал mc(рис. [4.1]).

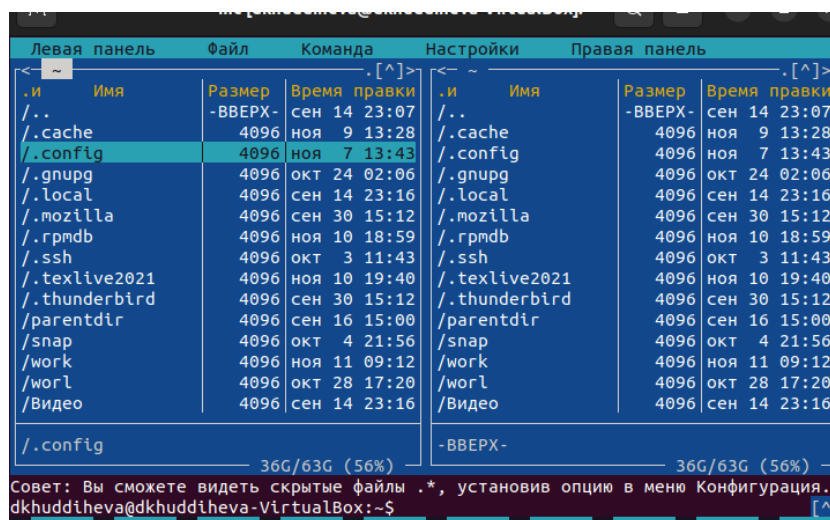


Рис. 4.1: Открытый mc

Перехожу в каталог ~/work/study/2023-2024/Архитектура компьютера/arch-рс,используя файловой менеджер mc (рис. [4.2]).

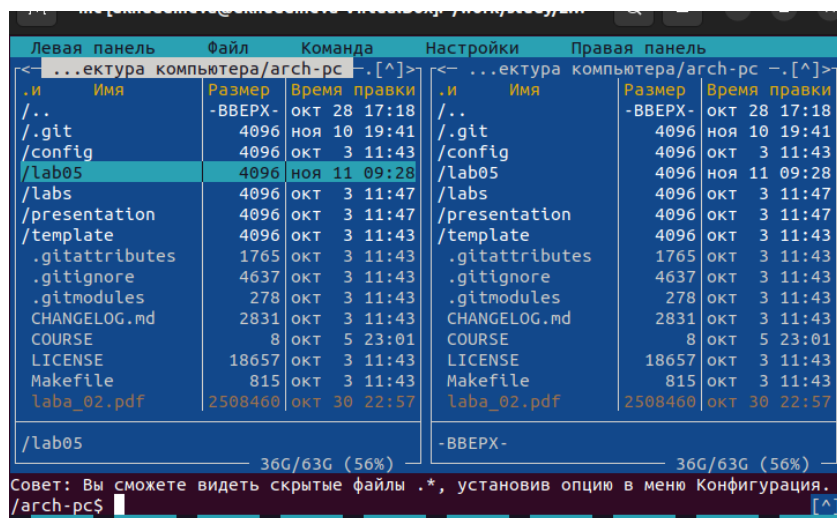


Рис. 4.2: Перемещение между директориями

С помощью функциональной клавиши F7 создаю каталог lab05 (рис. [4.3]).

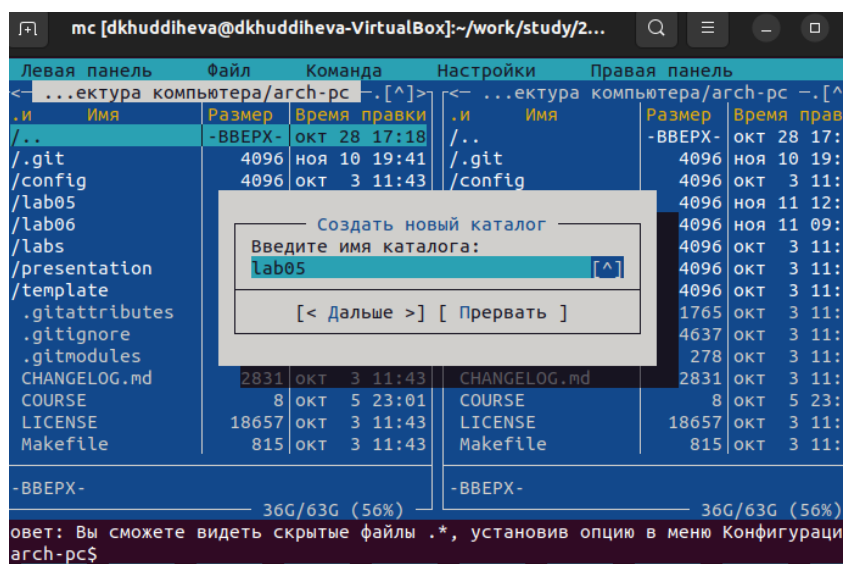


Рис. 4.3: Создание каталога

Перехожу в созданный каталог (рис. [4.4]).


```

GNU nano 6.2 /home/dkhuddiheva/work/study/2023-2024/Архитектура компь
SECTION .data ;Секция иницированных данных
msg: DB 'Введите строку:',10 ;сообщение плюс
; символ перевода строки
msgLen: EQU $-msg ;Длина переменной 'msg'

SECTION .bss ;Секция не иницированных данных
buf1: RESB 80 ;Буфер размером 80 байт

SECTION .txt ;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 '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 ;дескриптор файла 0-стандартный ввод
mov ecx, buf1 ;адрес буфера под вводимую строку
mov edx, 80 ;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. 4.5: Редактирование файла

С помощью функциональной клавиши F3 открываю файл для просмотра, чтобы проверить, содержит ли файл текст программы (рис. [??]).

```

/home/dkhuddiheva/work/study/2023-2024/Архитектура компьютера/arch-pc/lab05/lab5-1.asm 1614/1614
SECTION .data ;Секция иницированных данных
msg: DB 'Введите строку:',10 ;сообщение плюс
; символ перевода строки
msgLen: EQU $-msg ;Длина переменной 'msg'

SECTION .bss ;Секция не иницированных данных
buf1: RESB 80 ;Буфер размером 80 байт

SECTION .txt ;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 '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 ;дескриптор файла 0-стандартный ввод
mov ecx, buf1 ;адрес буфера под вводимую строку
mov edx, 80 ;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. 4.6: Открытие файла для просмотра

Транслирую текст программы файла в объектный файл командой `nasm -f elf lab5-1.asm`. Создался объектный файл `lab5-1.o`. Выполняю компоновку объектного

файла с помощью команды `ld -m elf_i386 -o lab5-1 lab5-1.o`. Создался исполняемый файл `lab5-1`. Запускаю исполняемый файл. Программа выводит строку “Введите строку:” и ждёт ввода с клавиатуры, я ввожу свои ФИО, на этом программа заканчивает свою работу (рис. [??]).

```

dkhuddiheva@dkhuddiheva-VirtualBox:~$ cd ~/work/study/2023-2024/"Архитектура компьютера"/arch-pc
dkhuddiheva@dkhuddiheva-VirtualBox:~/work/study/2023-2024/Архитектура компьютера/arch-pc$ mc
dkhuddiheva@dkhuddiheva-VirtualBox:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab05$ nasm -f elf lab5-1.asm
dkhuddiheva@dkhuddiheva-VirtualBox:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab05$ ld -m elf_i386 -o lab5-1 lab5-1.o
dkhuddiheva@dkhuddiheva-VirtualBox:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab05$ ./lab5-1
Введите строку:
Худдыева Дженет

```

4.2 Подключение внешнего файла

Скачиваю файл `in_out.asm` со страницы в ТУИС. Он сохранился в каталоге “Загрузки”. С помощью функциональной клавиши F5 копирую файл `in_out.asm` из каталоге Загрузки в созданный каталог `lab05` (рис. [4.7]).

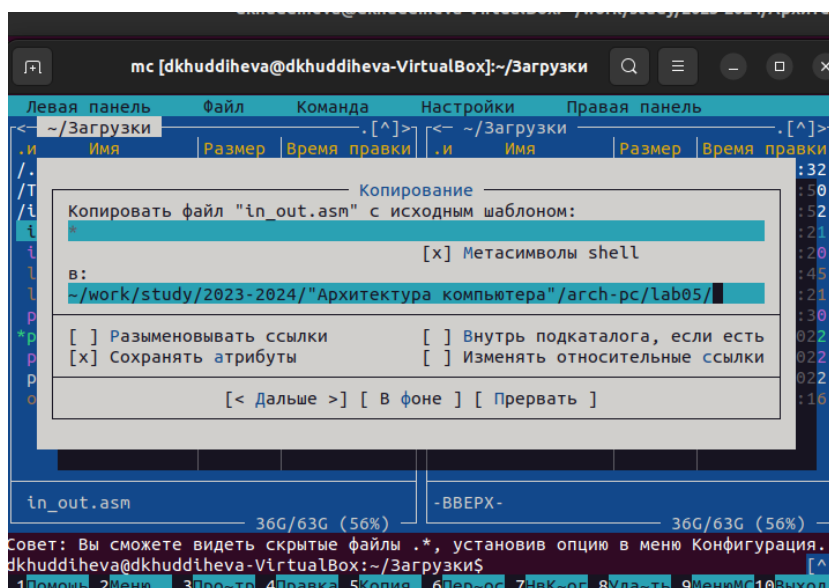


Рис. 4.7: Копирование файла

С помощью функциональной клавиши F5 копирую файл `lab5-1` в тот же ката-

лог,но с другим именем,для этого в появившемся окне mc прописываю имя для копии файла (рис. [4.8]).

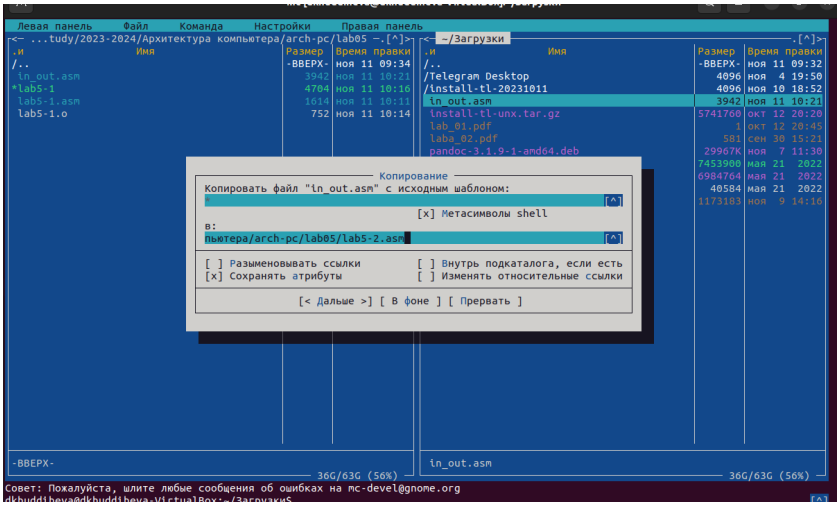


Рис. 4.8: Копирование файла

Изменяю содержимого файла lab5-2.asm во встроенном редакторе nano (рис. [4.9]).

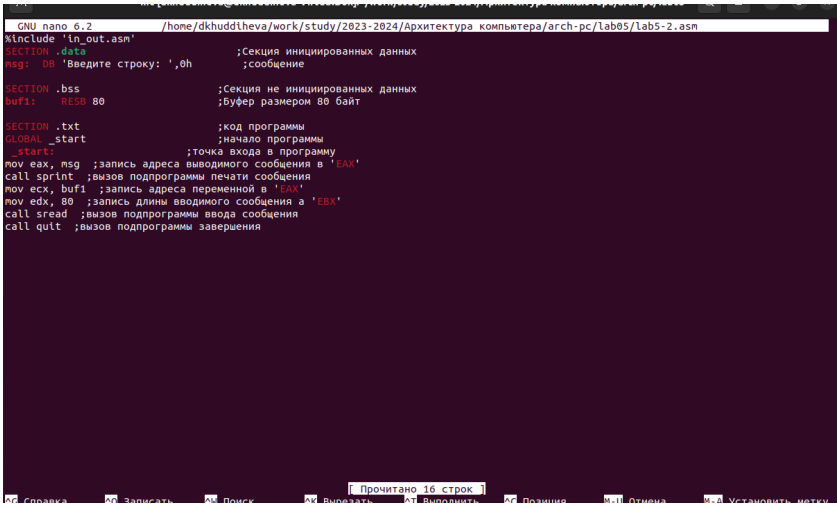


Рис. 4.9: Редактирование файла

Транслирую текст программы файла в объектный файл командой `nasm -f elf lab5-2.asm`.Создался объектный файл lab5-2.o.Выполняю компоновку объектного

файла с помощью команды `ld -m elf_i386 -o lab5-2 lab5-2.o`. Создался исполняемый файл `lab5-2` (рис. [4.10]).

```
dkhuddiheva@dkhuddiheva-VirtualBox:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab05$ nasm -f elf lab5-2.asm
dkhuddiheva@dkhuddiheva-VirtualBox:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab05$ ld -m elf_i386 -o lab5-2 lab5-2.o
dkhuddiheva@dkhuddiheva-VirtualBox:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab05$ ./lab5-2
Введите строку:
Худяева Дженнет
```

Рис. 4.10: Исполнение файла

Открывю файл `lab5-2.asm` для редактирования в nano функциональной клавишей F4. Изменяю в нём подпрограмму `sprintLF` на `sprint`. сохраняю файл для просмотра, чтобы проверить сохранение действий (рис. [4.11]).

```
mc [dkhuddiheva@dkhuddiheva-VirtualBox]:~/work/study/2... 1037/1037 1
/home/dkhuddiheva/work/~ch-pc/lab05/lab5-2.asm
%include 'in_out.asm'
SECTION .data
msg: DB 'Введите строку: ',0h
SECTION .bss
buf1: RESB 80
SECTION .txt
GLOBAL _start
_start:
mov eax, msg ;запись адреса выводимого сообщения в 'EAX'
call sprint ;вызов подпрограммы печати сообщения
mov ecx, buf1 ;запись адреса переменной в 'EAX'
mov edx, 80 ;запись длины вводимого сообщения в 'EDX'
call sread ;вызов подпрограммы ввода сообщения
call quit ;вызов подпрограммы завершения
```

Рис. 4.11: Отредактированный файл

Снова транслирую файл, выполняю компоновку созданного объектного файла, запускаю новый исполняемый файл (рис. [4.12]).

```
dkhuddiheva@dkhuddiheva-VirtualBox:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab05$ nasm -f elf lab5-2.asm
dkhuddiheva@dkhuddiheva-VirtualBox:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab05$ ld -m elf_i386 -o lab5-2 lab5-2.o
dkhuddiheva@dkhuddiheva-VirtualBox:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab05$ ./lab5-2
Введите строку: Худяева Дженнет
```

Рис. 4.12: Исполнение файла

4.3 Выполнение заданий для самостоятельной работы

1.Создаю копию файла lab5-1.asm с именем lab5-1-1.asm с помощью функциональной клавиши F5 (рис. [4.13]).

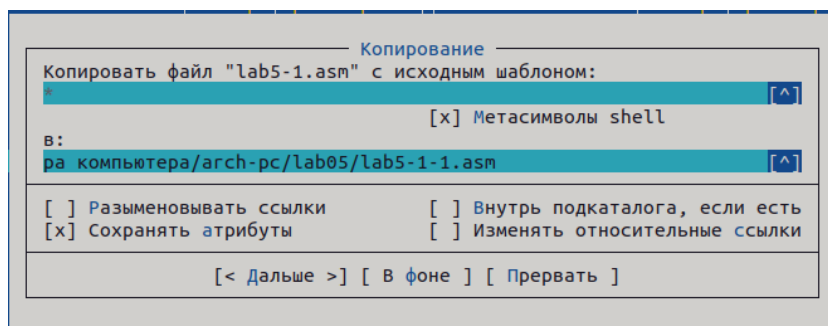


Рис. 4.13: Копирование файла

С помощью функциональной клавиши F4 открываю созданный файл для редактирования.Изменяю программу так,чтобы кроме вывода приглашения и запроса ввода,она выводила вводимую пользователем строку (рис. [4.14]).


```
GNU nano 6.2 /home/dkhuddiheva/work/study/2023-2024/Архитектура компьютера/arch-pc/lab05$
SECTION .data ;Секция инициализированных данных
msg: DB 'Введите строку:',10 ;сообщение плюс

msgLen: EQU $-msg ;длина переменной 'msg'

SECTION .bss ;Секция не инициализированных данных
buf1: RESB 80 ;Буфер размером 80 байт

SECTION .txt ;код программы
GLOBAL _start ;начало программы
_start: ;точка входа в программу
mov eax,4
mov ebx,1
mov ecx,msg
mov edx,msgLen
int 80h
mov eax, 3
mov ebx, 0
mov ecx, buf1
mov edx, 80
int 80h
mov eax,4
mov ebx,1
mov ecx,buf1
mov edx,buf1
int 80h
mov eax,1
mov ebx,0
int 80h
```

Рис. 4.14: Редактирование файла

2.Создаю объектный файл lab5-1-1.o,отдаю его на обработку компоновщику,получаю исполняемый файл.Программа запрашивает ввод, ввожу свои ФИО,далее программа выводит введенные мною данные (рис. [4.15]).

```
dkhuddiheva@dkhuddiheva-VirtualBox:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab05$ nasm -f elf lab5-1-1.asm
lab5-1-1.asm:23: error: parser: instruction expected
dkhuddiheva@dkhuddiheva-VirtualBox:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab05$ ld -n elf_386 -o lab5-1-1 lab5-1.o
dkhuddiheva@dkhuddiheva-VirtualBox:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab05$ ./lab5-1-1
Введите строку:
Худдыева Дженнет
```

Рис. 4.15: Исполнение файла

Код программы из пункта 1:

SECTION .data ;Секция инициализированных данных

msg: **DB** 'Введите строку:',10

```

msglen: EQU $-msg ;Длина переменной 'msg'
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,4
mov ebx,1
mov ecx,msg
mov edx,msgLen
int 80h

mov eax, 3
mov ebx, 0
mov ecx, buf1
mov edx, 80
int 80h

mov eax,4
mov ebx,1
mov ecx,buf1
mov edx,buf1
int 80h

mov eax,1
mov ebx,0
int 80h

```

3.Создаю копии файла lab5-2.asm с именем lab5-2-1.asm с помощью функциональной клавиши F5 (рис. [4.16]).

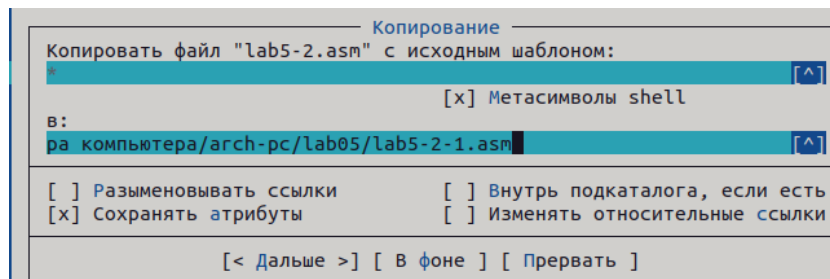


Рис. 4.16: Копирование файла

С помощью функциональной клавиши F4 открываю созданный файл для редактирования. Изменяю программу так чтобы кроме вывода приглашения и запроса ввода, она выводила вводимую пользователем строку (рис. [4.17]).

```
GNU nano 6.2 /home/dkhuddiheva/work/study/2023-2024/Архитект
#include 'in_out.asm'
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 .txt ;код программы
GLOBAL _start ;начало программы
_start: ;точка входа в программу
mov eax, msg ;запись адреса выводимого сообщения в 'EAX'
call sprint ;вызов подпрограммы печати сообщения
mov ecx, buf1 ;запись адреса переменной в 'EAX'
mov edx, 80 ;запись длины вводимого сообщения а 'EBX'
call sread ;вызов подпрограммы ввода сообщения
mov eax, 4
mov ebx, 1
mov ecx, buf1
int 80h
call quit ;вызов подпрограммы завершения
```

Рис. 4.17: Редактирование файла

4. Создаю объектный файл lab5-2-1.o, отдаю его на обработку компоновщи- ку, получаю исполняемый файл lab5-2-1, запускаю полученный исполняемый файл. Программа запрашивает ввод без переноса на новую строку, ввожу свои ФИО, далее программа выводит введенные мною данные (рис. [4.18]).

```
dkhuddthevagdkhuddtheva-VirtualBox: ~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab05$ nasm -f elf lab5-2-1.asm
dkhuddthevagdkhuddtheva-VirtualBox: ~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab05$ ld -m elf_i386 -o lab5-2-1 lab5-2-1.o
dkhuddthevagdkhuddtheva-VirtualBox: ~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab05$ ./lab5-2-1
Введите строку: Худыева Дженнет
```

Рис. 4.18: Исполнение файла

5 Выводы

При выполнении данной лабораторной работы я приобрела практические навыки работы в Midnight Commander, а так же освоила инструкции языка ассемблера `mov` и `int`