

Отчёт по лабораторной работе №5

Дисциплина: архитектура компьютера

Юлдошев Давлатджон Шухратович

Содержание

1	Цель работы	1
2	Выполнение лабораторной работы.....	1
3	Выполнение заданий для самостоятельной работы.....	5
3.1	Задание №1.....	5
3.2	Задание №2.....	7
4	Вывод.....	8

1 Цель работы

Цель данной лабораторной работы - приобретение практических навыков работы в Midnight Commander и освоение инструкций языка ассемблера mov и int

2 Выполнение лабораторной работы

Шаг 1

Открываю Midnight Commander, перехожу в каталог arch-pc и создаю рабочий каталог lab05 (рис. [1]).

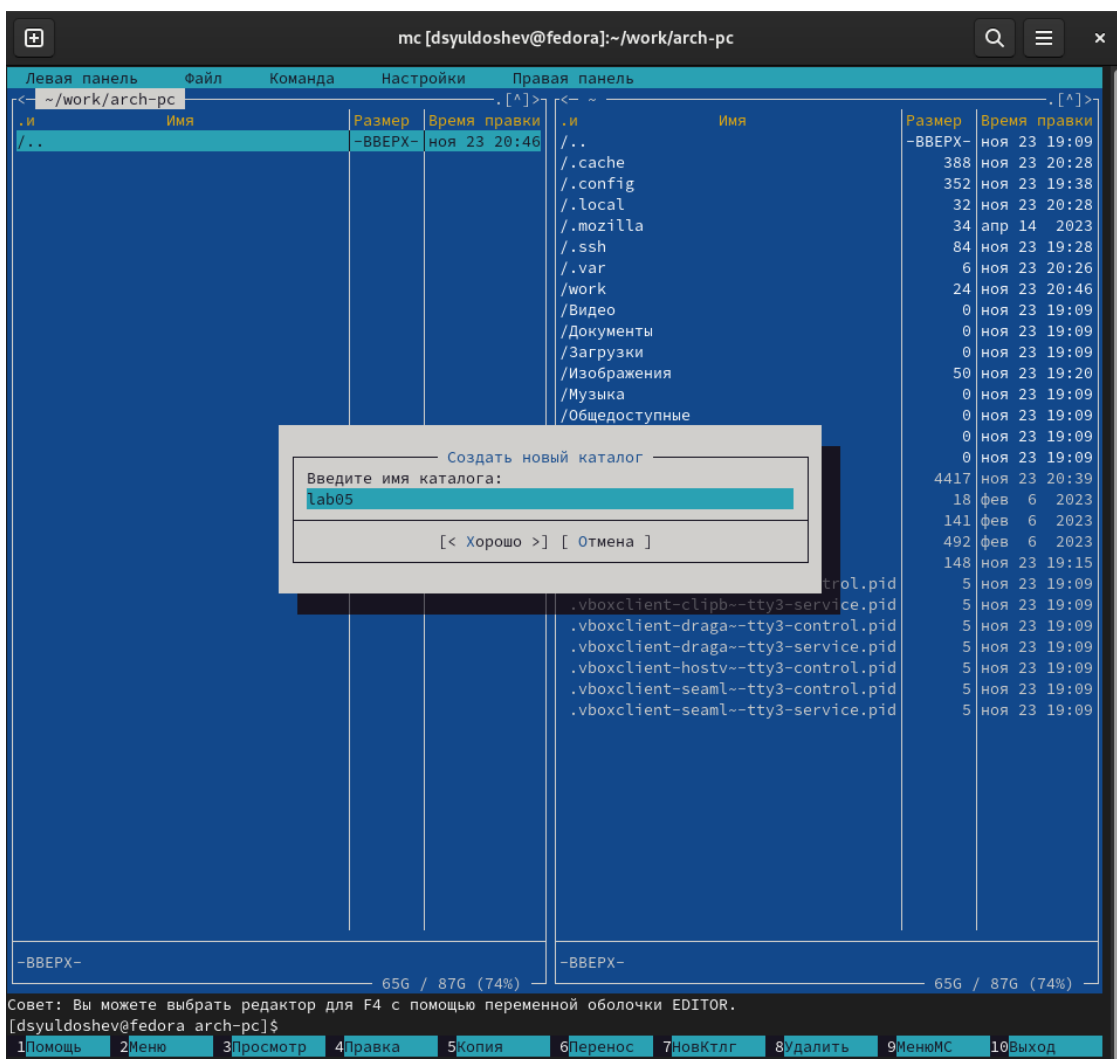


Figure 1: Midnight Commander

Шаг 2

Перехожу в созданную директорию и создаю файл lab5-1.asm (рис. [2]).

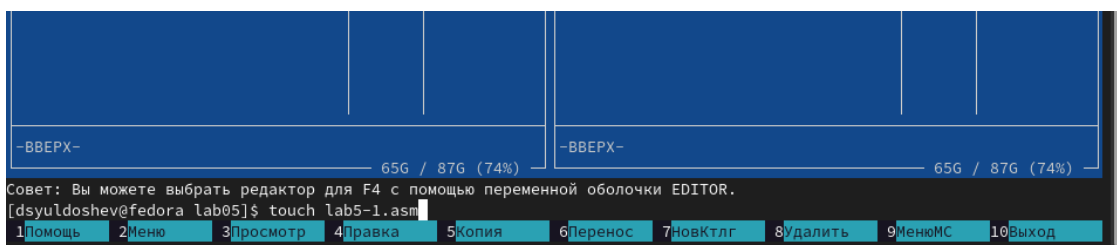
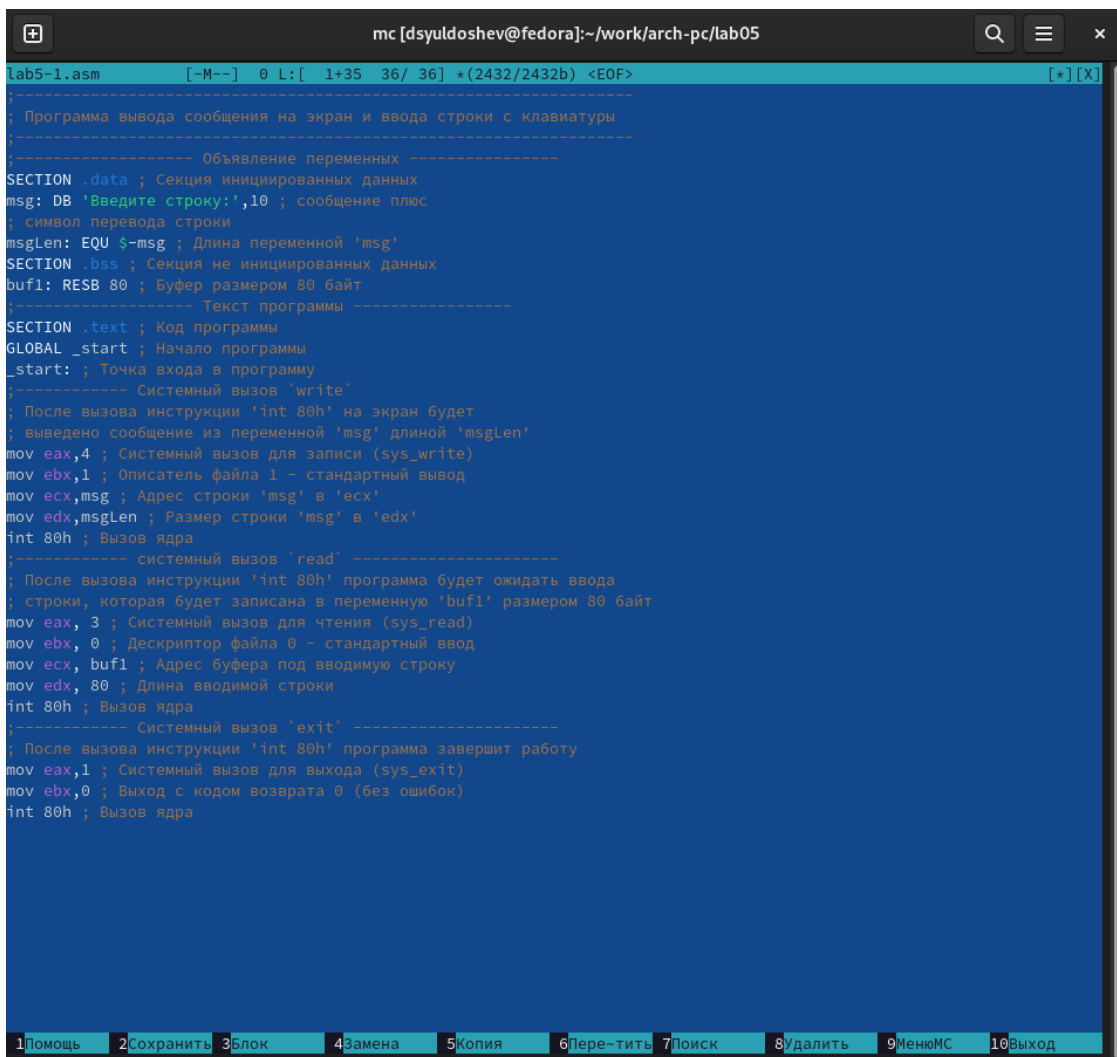


Figure 2: Создание файла

Шаг 3

Открываю файл lab5-1.asm для редактирования во встроенном редакторе nano и вставляю в него программу (рис. [3]).



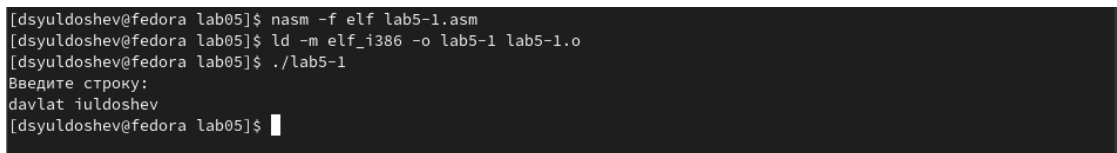
```
mc [dsyuldoshev@fedora]::~/work/arch-pc/lab05
lab5-1.asm [-M--] 0 L: [ 1+35 36/ 36] *(2432/2432b) <EOF> [*] [X]
; Программа вывода сообщения на экран и ввода строки с клавиатуры
;----- Объявление переменных -----
SECTION .data ; Секция инициализированных данных
msg: DB 'Введите строку:',10 ; сообщение плюс
; символ перевода строки
msgLen: EQU $-msg ; Длина переменной 'msg'
SECTION .bss ; Секция не инициализированных данных
buf1: RESB 80 ; Буфер размером 80 байт
;----- Текст программы -----
SECTION .text ; Код программы
GLOBAL _start ; Начало программы
_start: ; Точка входа в программу
;----- Системный вызов 'write' -----
; После вызова инструкции 'int 80h' на экран будет
; выведено сообщение из переменной 'msg' длиной 'msgLen'
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
;----- системный вызов 'read' -----
; После вызова инструкции 'int 80h' программа будет ожидать ввода
; строки, которая будет записана в переменную 'buf1' размером 80 байт
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
;----- Системный вызов 'exit' -----
; После вызова инструкции 'int 80h' программа завершит работу
mov eax,1 ; Системный вызов для выхода (sys_exit)
mov ebx,0 ; Выход с кодом возврата 0 (без ошибок)
int 80h ; Вызов ядра

1Помощь 2Сохранить 3Блок 4Замена 5Копия 6Пере-тить 7Поиск 8Удалить 9МенюМС 10Выход
```

Figure 3: Изменение файла

Шаг 4

Создаю объектный файл lab5-1.o, выполняю компоновку объектного файла и запускаю исполняемый файл (рис. [4]).



```
[dsyuldoshev@fedora lab05]$ nasm -f elf lab5-1.asm
[dsyuldoshev@fedora lab05]$ ld -m elf_i386 -o lab5-1 lab5-1.o
[dsyuldoshev@fedora lab05]$ ./lab5-1
Введите строку:
davlat iuldoshev
[dsyuldoshev@fedora lab05]$
```

Figure 4: Запуск программы

Шаг 6

Скачиваю файл in_out.asm, копирую его из директории Загрузки в директорию lab05 (рис. [5]).

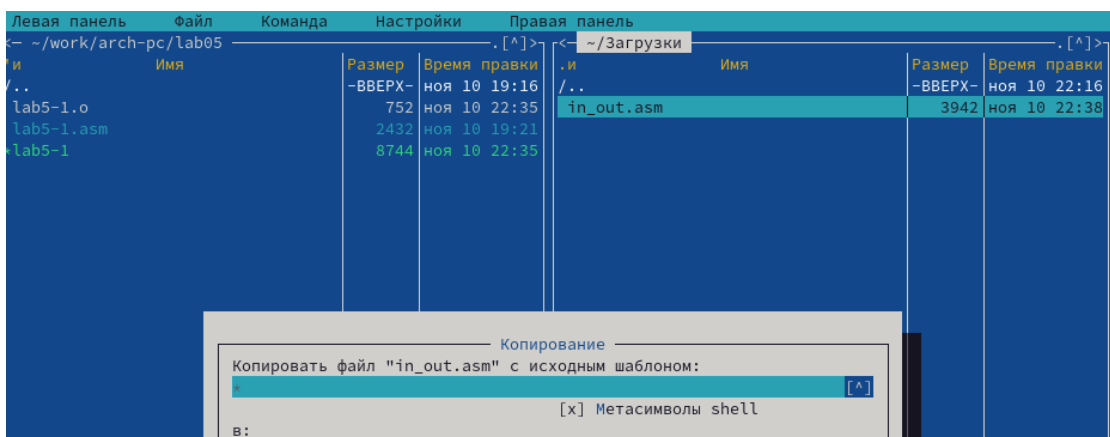


Figure 5: Копирование файла

Шаг 7

Создаю файл lab5-1.asm с новым именем lab5-2.asm (рис. [6]).

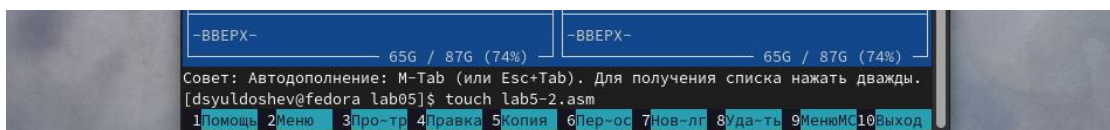


Figure 6: Копирование файла

Шаг 8

Изменяю текст программы lab5-2.asm, чтобы в ней использовались функции из подключаемого файла (рис. [7]).

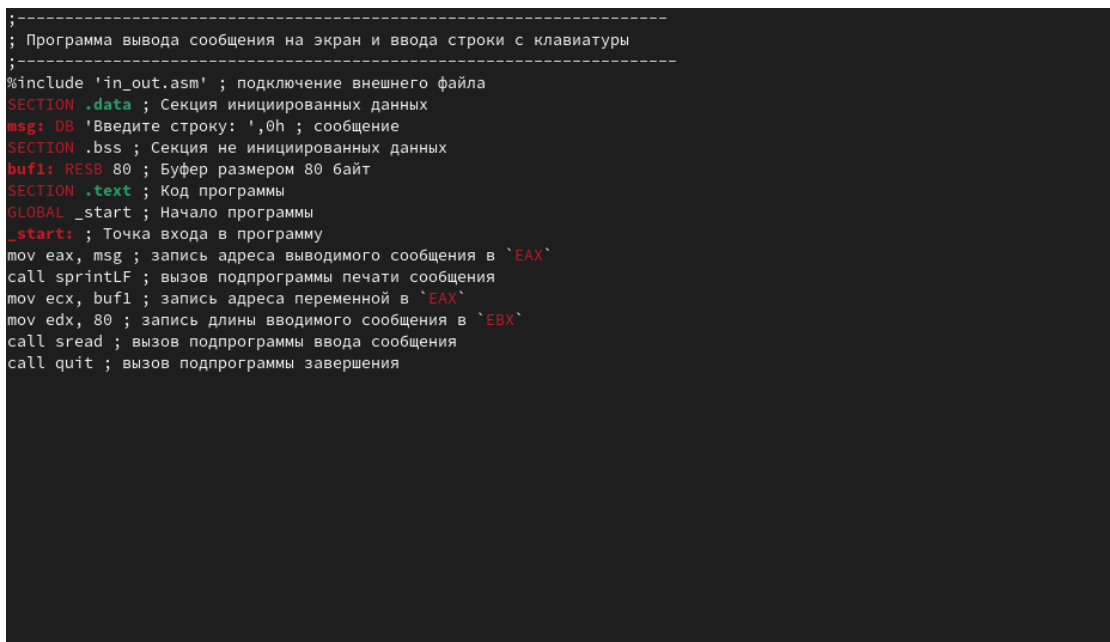


Figure 7: Изменение программы

Шаг 9

Создаю объектный файл lab5-2.o, выполняю компоновку объектного файла и запускаю исполняемый файл (рис. [8]).

```
[dsyuldoshev@fedora lab05]$ nasm -f elf lab5-2.asm
[dsyuldoshev@fedora lab05]$ ld -m elf_i386 -o lab5-2 lab5-2.o
[dsyuldoshev@fedora lab05]$ ./lab5-2
Введите строку: ^C
[dsyuldoshev@fedora lab05]$ ./lab5-2
Введите строку: давлат юлдошев
[dsyuldoshev@fedora lab05]$
```

Figure 8: Запуск программы

Шаг 10

В тексте программы lab5-2.asm заменяю *sprintLF* на *sprint* и Создаю объектный файл lab5-2.o, выполняю компоновку объектного файла и запускаю исполняемый файл (рис. [9]).

```
[dsyuldoshev@fedora lab05]$ nano lab5-2.asm
[dsyuldoshev@fedora lab05]$ nasm -f elf lab5-2.asm
[dsyuldoshev@fedora lab05]$ ld -m elf_i386 -o lab5-2 lab5-2.o
[dsyuldoshev@fedora lab05]$ ./lab5-2
Введите строку:
Давлат Юлдошев
[dsyuldoshev@fedora lab05]$
```

Figure 9: Запуск программы

- Теперь ввод производится на той же строке, что и вывод, убран символ перевода строки после вывода.

3 Выполнение заданий для самостоятельной работы

3.1 Задание №1

1. Копирую файл lab5-1.asm с именем lab5-4.asm и Изменяю код программы, добавляя вывод введенной строки (рис. [10]).

```

dsyuldoshev@fedora:~/work/arch... x dsyuldoshev@fedora:~/work/arch... x
GNU nano 7.2 lab5-4.asm Изменён
;----- Объявление переменных -----
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
;----- Текст программы -----
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 '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

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

```

Figure 10: Изменение программы

- Текст программы из фотоотчёта (рис. [10]).

```

;----- Объявление переменных -----
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
;----- Текст программы -----
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 '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
;-----вывод на экран-----
int 80h ; Вызов ядра
mov eax,4 ; Системный вызов для записи (sys_write)
mov ebx,1 ; Описатель файла '1' - стандартный вывод
mov ecx,buf1 ; Адрес строки buf1 в ecx

```

```

mov edx,buf1 ; Размер строки buf1
;-----
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

```

3. Создаю объектный файл lab5-4.o, komponую его в исполняемый файл, запускаю исполняемый файл (рис. [11]).

```

[dsyuldoshev@fedora lab05]$ touch lab5-4.asm
[dsyuldoshev@fedora lab05]$ nano lab5-4.asm
[dsyuldoshev@fedora lab05]$ nasm -f elf lab5-4.asm
[dsyuldoshev@fedora lab05]$ ld -m elf_i386 -o lab5-4 lab5-4.o
[dsyuldoshev@fedora lab05]$ ./lab5-4
Введите строку:
Давлат Юлдошев
Давлат Юлдошев
[dsyuldoshev@fedora lab05]$

```

Figure 11: Запуск программы

3.2 Задание №2

1. Копирую файл lab5-2.asm с именем lab5-3.asm и Изменяю код программы, добавляя вывод введенной строки (рис. [12]).

```

dsyuldoshev@fedora:~/work/arch... x dsyuldoshev@fedora:~/work/arch... x
GNU nano 7.2 lab5-3.asm
;-----
; Программа вывода сообщения на экран и ввода строки с клавиатуры
;-----
#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 подпрограммы печати сообщения
;-----вывод на экран-----
mov ecx, buf1 ; запись адреса переменной в `EAX`
mov edx, 80 ; запись длины вводимого сообщения в `EBX`
call sread ; вызов подпрограммы ввода сообщения
mov eax,4 ; Системный вызов для записи (sys_write)
mov ebx,1 ; Описатель файла '1' - стандартный вывод
mov ecx,buf1 ; Адрес строки buf1 в ecx
[ Прочитано 22 строки ]
^G Справка ^O Записать ^W Поиск ^K Вырезать ^T Выполнить ^C Позиция
^X Выход ^R ЧитФайл ^\ Замена ^U Вставить ^J Выводить ^/_ К строке

```

Figure 12: Изменение файла

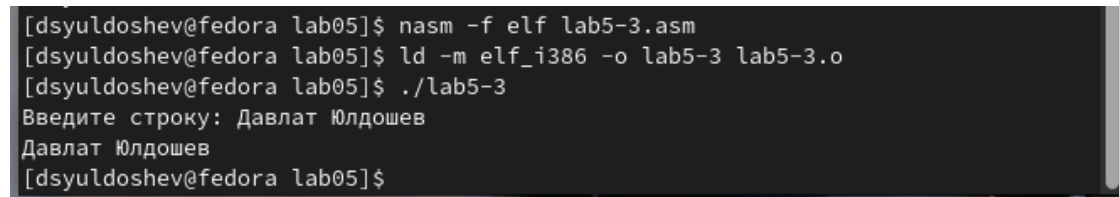
- Текст программы из фотоотчёта (рис. [12])

```

;-----
; Программа вывода сообщения на экран и ввода строки с клавиатуры
;-----
#include 'in_out.asm' ; подключение внешнего файла
SECTION .data ; Секция инициализированных данных
msg: DB 'Введите строку: ',0h ; сообщение
SECTION .bss ; Секция не инициализированных 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 подпрограммы печати сообщения
;-----вывод на экран-----
mov ecx, buf1 ; запись адреса переменной в `EAX`
mov edx, 80 ; запись длины вводимого сообщения в `EBX`
call sread ; вызов подпрограммы ввода сообщения
mov eax,4 ; Системный вызов для записи (sys_write)
mov ebx,1 ; Описатель файла '1' - стандартный вывод
mov ecx,buf1 ; Адрес строки buf1 в ecx
int 80h ; Вызов ядра
call quit ; вызов подпрограммы завершения

```

3. Создаю объектный файл lab5-4.o, компоную его в исполняемый файл, запускаю исполняемый файл (рис. [13]).



```

[dsyuldoshev@fedora lab05]$ nasm -f elf lab5-3.asm
[dsyuldoshev@fedora lab05]$ ld -m elf_i386 -o lab5-3 lab5-3.o
[dsyuldoshev@fedora lab05]$ ./lab5-3
Введите строку: Давлат Юлдошев
Давлат Юлдошев
[dsyuldoshev@fedora lab05]$

```

Figure 13: Запуск программы

4 Вывод

Я приобрел практические навыки работы в Midnight Commander и освоил инструкции mov и int в языке ассемблер.