

# **Отчет по лабораторной работе №5**

**Дисциплина: Архитектура компьютера**

Авдеенко Марьяна Дмитриевна

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Задание</b>	<b>6</b>
<b>3</b>	<b>Теоретическое введение</b>	<b>7</b>
<b>4</b>	<b>Выполнение лабораторной работы</b>	<b>8</b>
<b>5</b>	<b>Задания самостоятельной работы</b>	<b>18</b>
<b>6</b>	<b>Выводы</b>	<b>20</b>
	<b>Список литературы</b>	<b>21</b>

## Список иллюстраций

4.1	Каталог ~/work . . . . .	8
4.2	Каталог ~/work/arch-pc . . . . .	9
4.3	Создание папки . . . . .	9
4.4	Созданный каталог lab05 . . . . .	10
4.5	Созданный файл . . . . .	10
4.6	Файл для редактирования . . . . .	11
4.7	Отредактированный файл . . . . .	12
4.8	Исполняемый файл . . . . .	12
4.9	Скачанный файл in_out.asm . . . . .	13
4.10	Скачанный файл in_out.asm в папке lab05 . . . . .	14
4.11	Создание файла lab5-2.asm . . . . .	15
4.12	Изменения в тексте файла lab5-2.asm . . . . .	16
4.13	Замена подпрограммы sprintLF на sprint . . . . .	17
4.14	Проверка работы файла . . . . .	17
5.1	Файл lab5-3.asm . . . . .	18
5.2	Файл lab5-3.asm с выводом фамилии . . . . .	18
5.3	Файл lab5-4.asm . . . . .	19
5.4	Файл lab5-4.asm с выводом номера группы . . . . .	19

## Список таблиц

# 1 Цель работы

Приобрести практические навыки работы в Midnight Commander.

## 2 Задание

Здесь приводится описание задания в соответствии с рекомендациями методического пособия и выданным вариантом.

### 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.

Следующие комбинации клавиш облегчают работу с Midnight Commander: \* Tab используется для переключения между панелями; \* клавиши “вверх” и “вниз” используются для навигации, Enter для входа в каталог или открытия файла (если в файле расширений mc.ext заданы правила связи определённых расширений файлов с инструментами их запуска или обработки); \* Ctrl + u (или через меню Команда > Переставить панели) меняет местами содержимое правой и левой панелей; \* Ctrl + o (или через меню Команда > Отключить панели) скрывает или возвращает панели Midnight Commander, за которыми доступен для работы командный интерпретатор оболочки и выводимая туда информация. \* Ctrl + x + d (или через меню Команда > Сравнить каталоги) позволяет сравнить содержимое каталогов, отображаемых на левой и правой панелях. Дополнительную информацию о Midnight Commander можно получить по команде man mc и на странице проекта.

## 4 Выполнение лабораторной работы

- 1) Открыла *Midnight Commander*.
- 2) Пользуясь клавишами “вверх”, “вниз” и Enter перешла в каталог ~/work/arch-рс, созданный при выполнении лабораторной работы №4 (рис. 4.1, 4.2).

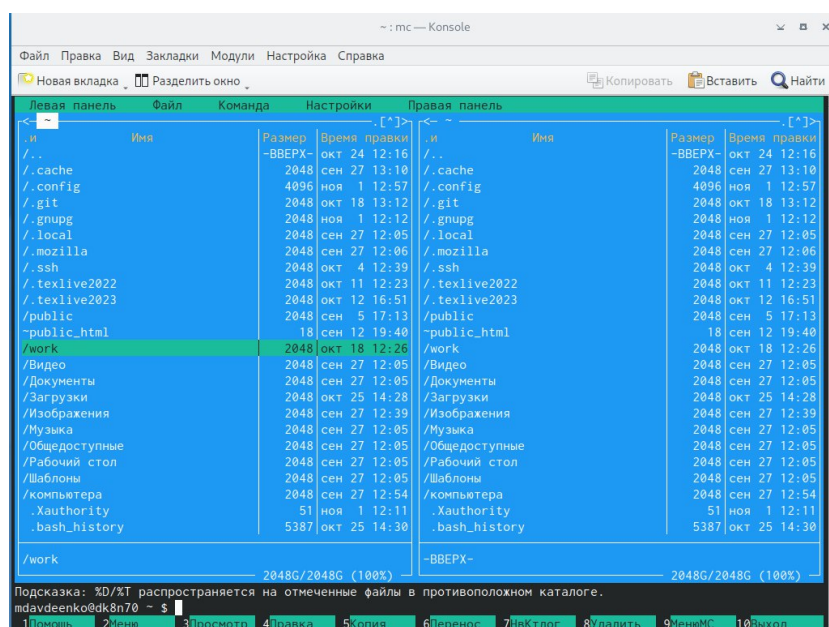


Рис. 4.1: Каталог ~/work



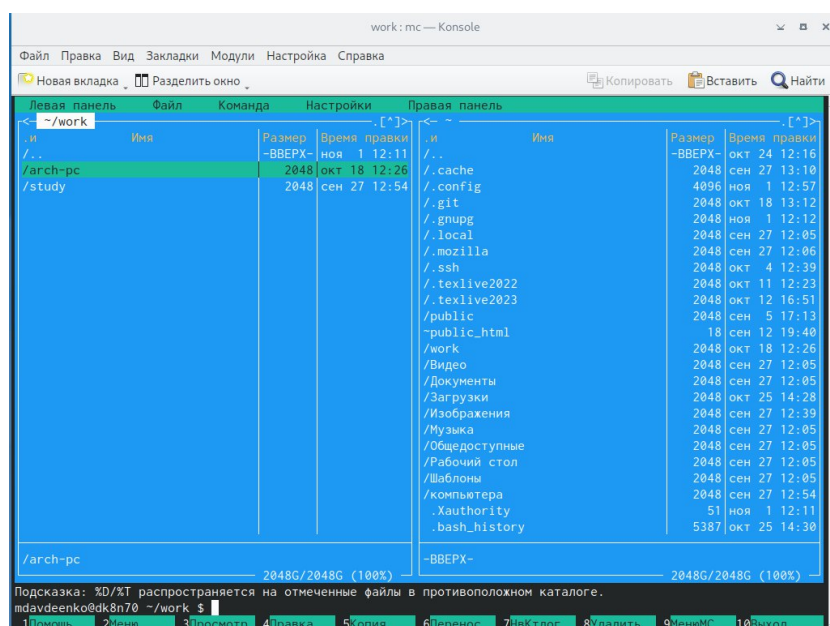


Рис. 4.2: Каталог ~/work/arch-pc

- 3) С помощью функциональной клавиши F7 создала папку lab05 (рис. 4.3) и перешла в созданный каталог (рис. 4.4).

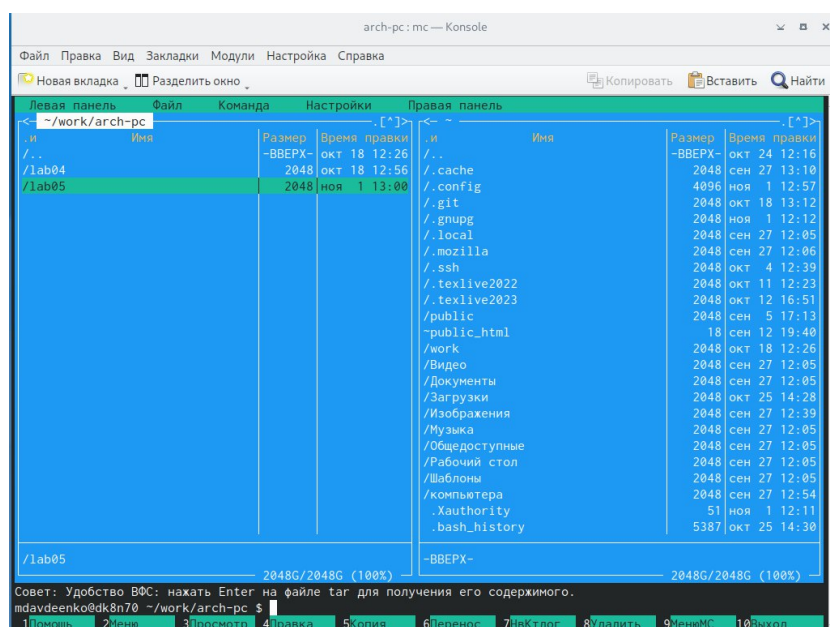


Рис. 4.3: Создание папки

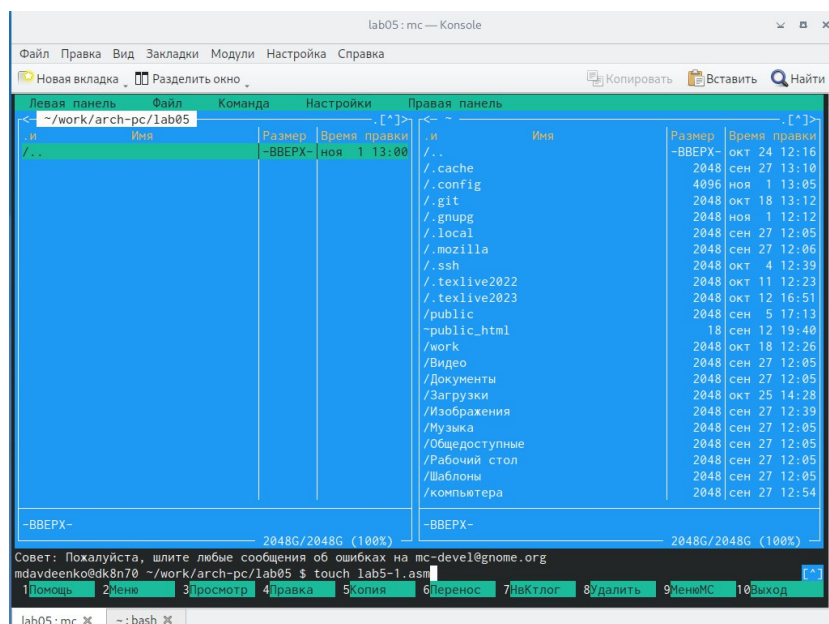


Рис. 4.4: Созданный каталог lab05

- 4) Пользуясь строкой ввода и командой touch создала файл lab5-1.asm (рис. 4.5).

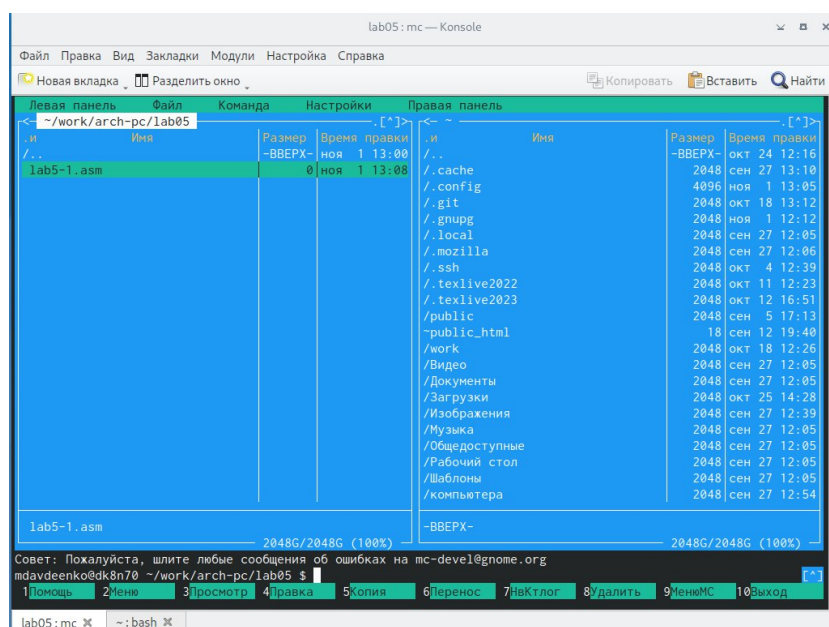
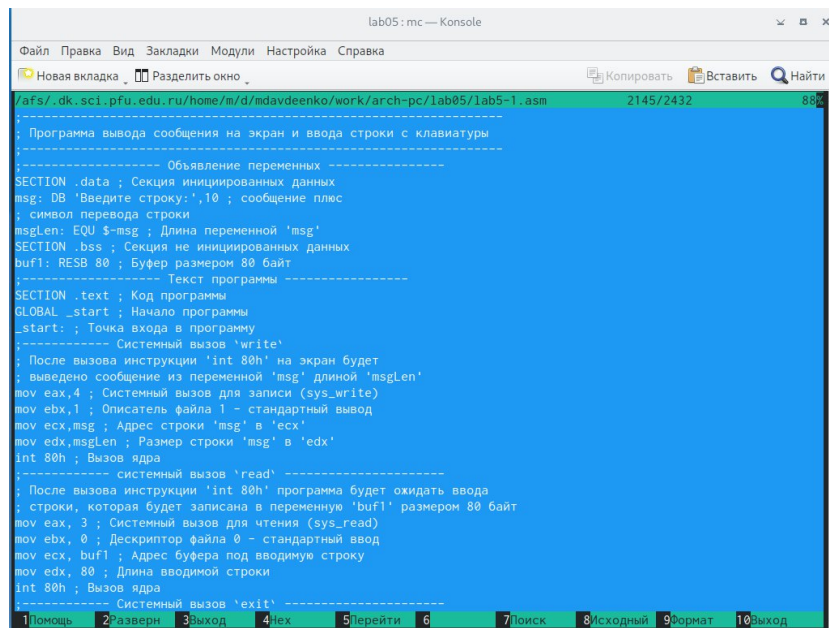


Рис. 4.5: Созданный файл

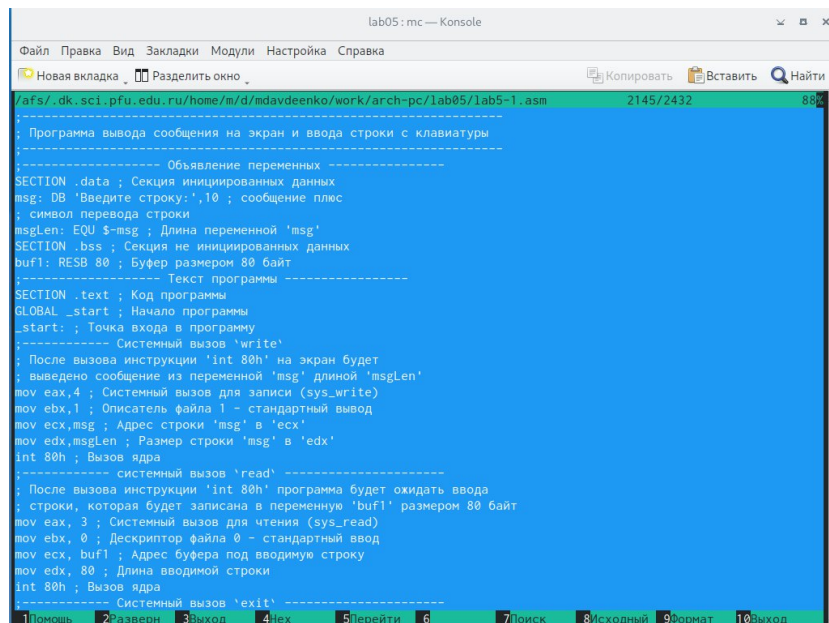
- 5) С помощью функциональной клавиши F4 открыла файл lab5-1.asm для редактирования во встроенном редакторе (рис. 4.6).



```
lab05: mc — Konsole
Файл Правка Вид Закладки Модули Настройка Справка
Новая вкладка Разделить окно Копировать Вставить Найти
/afs/.dk.sci.pfu.edu.ru/home/m/d/mdavdeenko/work/arch-pc/lab05/lab5-1.asm 2145/2432 88%
; Программа вывода сообщения на экран и ввода строки с клавиатуры
;----- 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
;----- 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' -----
;-----
```

Рис. 4.6: Файл для редактирования

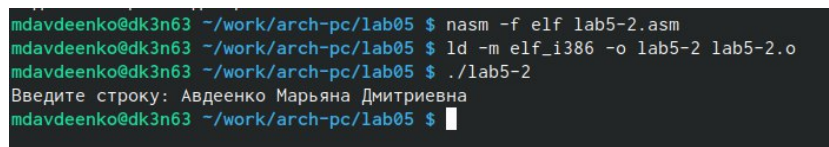
- 6) Ввела текст программы из листинга 5.1, сохранила изменения и закрыла файл.
- 7) С помощью функциональной клавиши F3 открыла файл lab5-1.asm для просмотра. Убедилась, что файл содержит текст программы. (рис. 4.7).



```
lab05: mc — Konsole
Файл  Правка  Вид  Закладки  Модули  Настройка  Справка
+ Новая вкладка  + Разделить окно  + Копировать  + Вставить  + Найти
/afs/.dk.sci.pfu.edu.ru/home/m/dmdavdeenko/work/arch-pc/lab05/lab5-1.asm  2145/2432  88%
-----
; Программа вывода сообщения на экран и ввода строки с клавиатуры
-----
; 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
;----- 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 ;Descriptor 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'
;-----
1Помощь 2Разверн 3Выход 4Чех 5Перейти 6 7Поиск 8Исходный 9Формат 10Выход
```

Рис. 4.7: Отредактированный файл

- 8) Оттранслировала текст программы lab5-1.asm в объектный файл. Выполнила компоновку объектного файла и запустила получившийся исполняемый файл. Программа вывела строку 'Введите строку:', где ввела свои ФИО (рис. 4.8).



```
mdavdeenko@dk3n63 ~/work/arch-pc/lab05 $ nasm -f elf lab5-2.asm
mdavdeenko@dk3n63 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-2 lab5-2.o
mdavdeenko@dk3n63 ~/work/arch-pc/lab05 $ ./lab5-2
Введите строку: Авдеенко Марьяна Дмитриевна
mdavdeenko@dk3n63 ~/work/arch-pc/lab05 $
```

Рис. 4.8: Исполняемый файл

- 9) Скачала файл in\_out.asm со страницы курса в ТУИС (рис. 4.9).

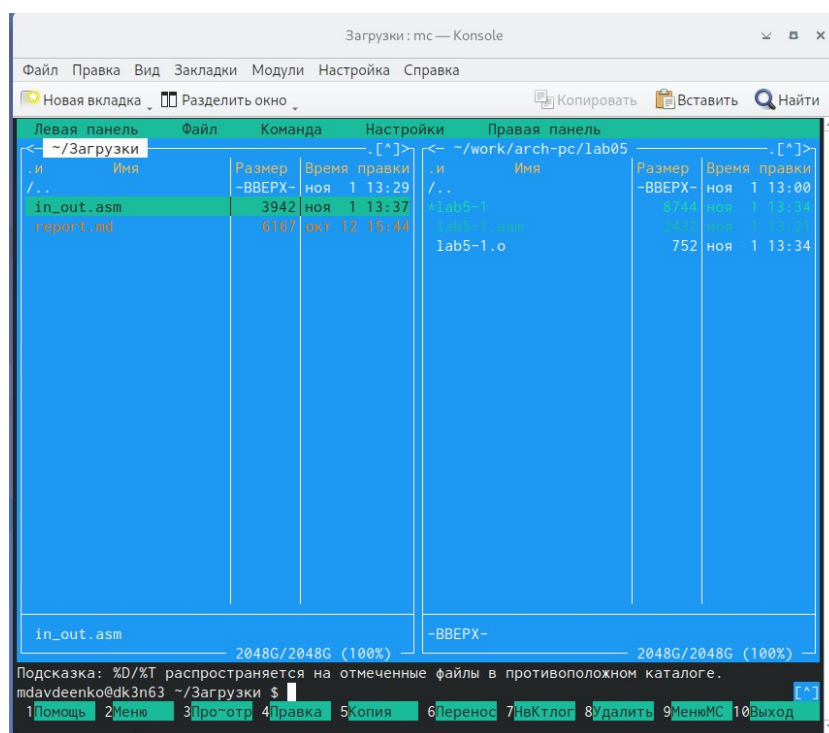


Рис. 4.9: Скачанный файл in\_out.asm

- 10) Подключаемый файл in\_out.asm должен лежать в том же каталоге, что и файл с программой, в которой он используется, поэтому файл in\_out.asm переложила в папку work/arch-pc/lab05 (рис. 4.10).

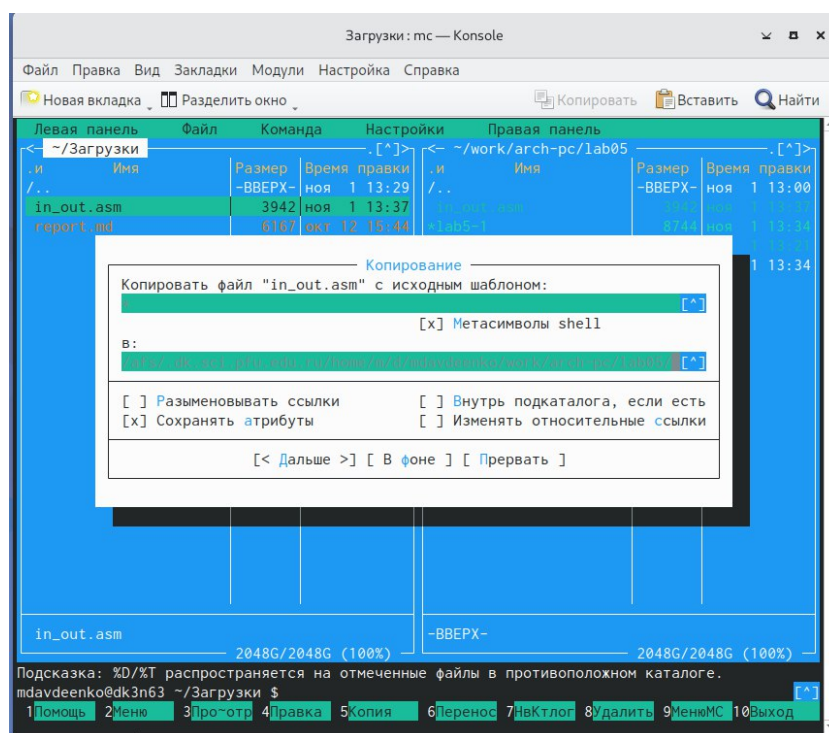


Рис. 4.10: Скачанный файл in\_out.asm в папке lab05

- 11) С помощью функциональной клавиши F6 создала копию файла lab5-1.asm с именем lab5-2.asm. Выделила файл lab5-1.asm, нажала клавишу F6, ввела имя файла lab5-2.asm и нажала клавишу Enter (рис. 4.11).



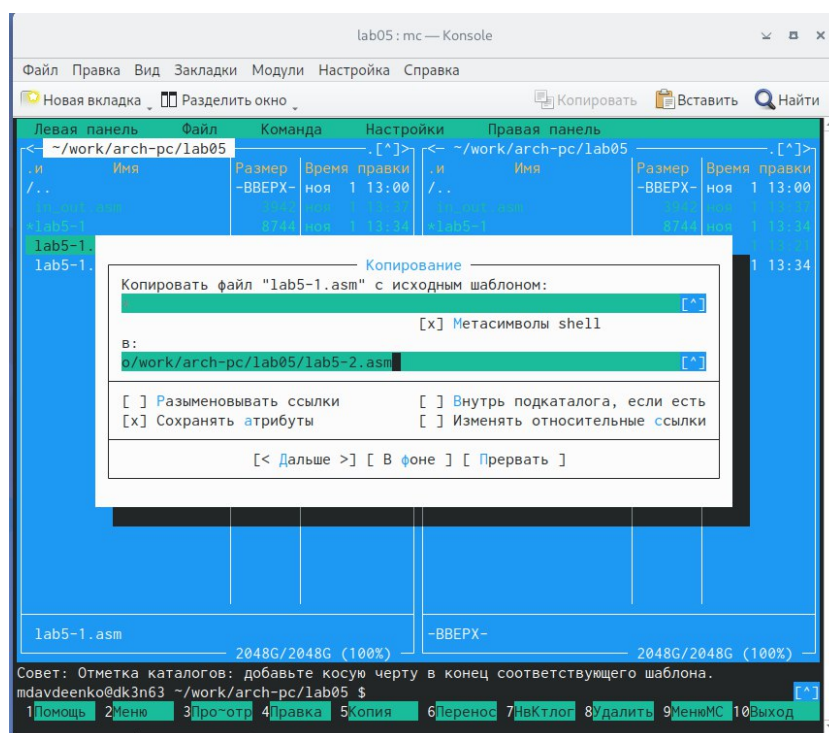
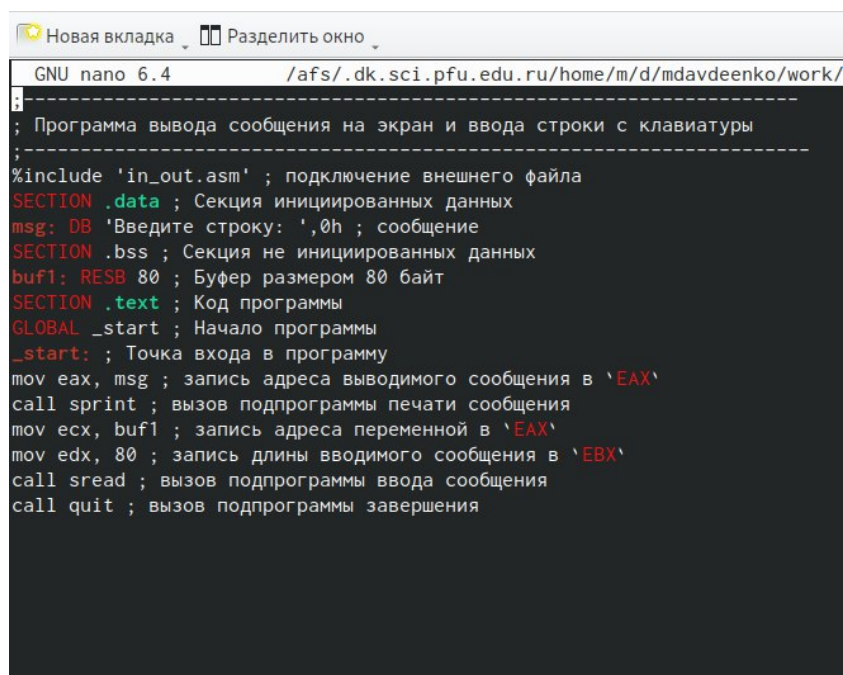


Рис. 4.11: Создание файла lab5-2.asm

- 12) Исправила текст программы в файле lab5-2.asm с использованием подпрограмм из внешнего файла in\_out.asm в соответствии с листингом 5.2 (рис. 4.12). Создала исполняемый файл и проверила его работу.

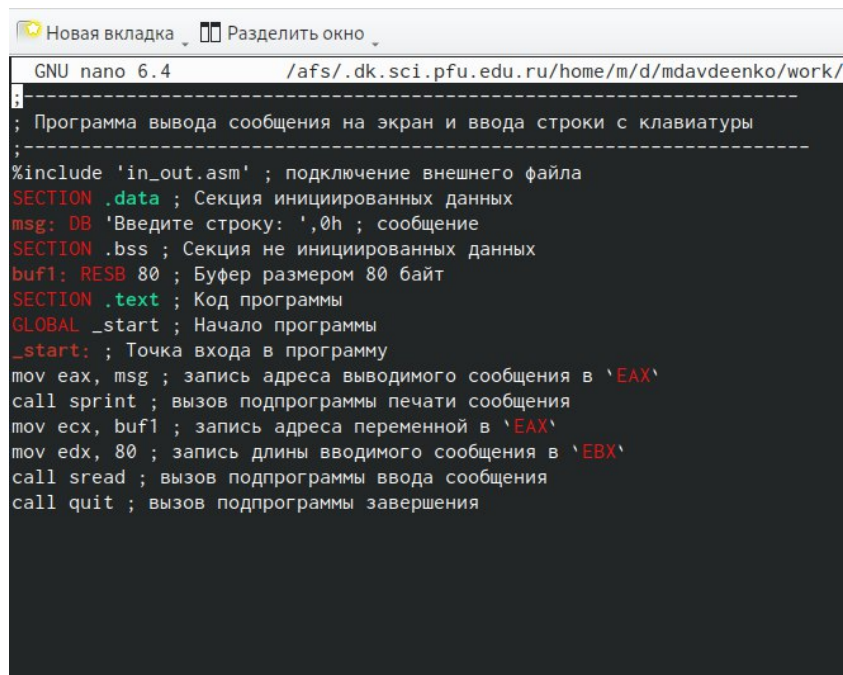


```
GNU nano 6.4 /afs/.dk.sci.pfu.edu.ru/home/m/d/mdavdeenko/work/
;-----
; Программа вывода сообщения на экран и ввода строки с клавиатуры
;-----
#include 'in_out.asm' ; подключение внешнего файла
SECTION .data ; Секция инициированных данных
msg: DB 'Введите строку: ',0h ; сообщение
SECTION .bss ; Секция 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
mov ecx, buf1 ; запись адреса переменной в 'EAX'
mov edx, 80 ; запись длины вводимого сообщения в 'EBX'
call sread ; вызов подпрограммы ввода сообщения
call quit ; вызов подпрограммы завершения
```

Рис. 4.12: Изменения в тексте файла lab5-2.asm

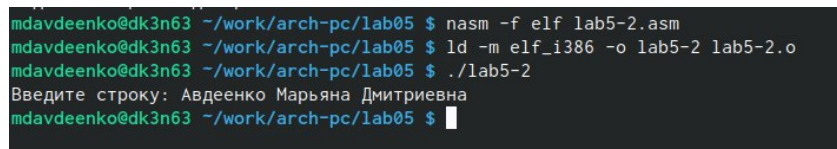
- 13) В файле lab5-2.asm заменила подпрограмму sprintLF на sprint (рис. 4.13).  
Создала исполняемый файл и проверьте его работу (рис. 4.14).





```
GNU nano 6.4 /afs/.dk.sci.pfu.edu.ru/home/m/d/mdavdeenko/work/
;-----
; 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
mov ecx, buf1 ; запись адреса переменной в 'EAX'
mov edx, 80 ; запись длины вводимого сообщения в 'EBX'
call sread ; вызов подпрограммы ввода сообщения
call quit ; вызов подпрограммы завершения
```

Рис. 4.13: Замена подпрограммы sprintLF на sprint



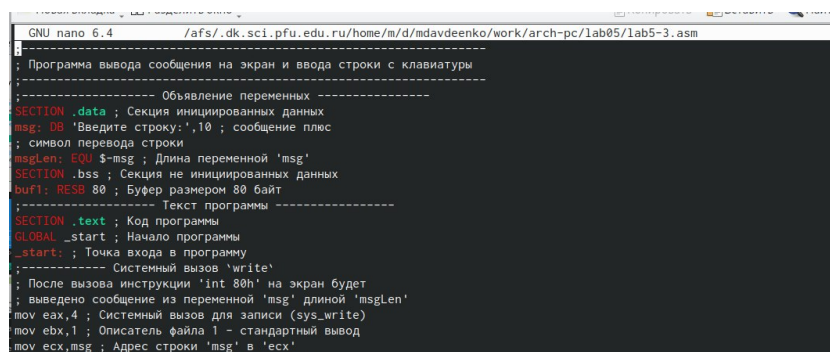
```
mdavdeenko@dk3n63 ~/work/arch-pc/lab05 $ nasm -f elf lab5-2.asm
mdavdeenko@dk3n63 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-2 lab5-2.o
mdavdeenko@dk3n63 ~/work/arch-pc/lab05 $ ./lab5-2
Введите строку: Авдеенко Марьяна Дмитриевна
mdavdeenko@dk3n63 ~/work/arch-pc/lab05 $
```

Рис. 4.14: Проверка работы файла

## 5 Задания самостоятельной работы

1) Создала копию файла lab5-1.asm. Внесла изменения в программу (без использования внешнего файла in\_out.asm), так чтобы она работала по следующему алгоритму:

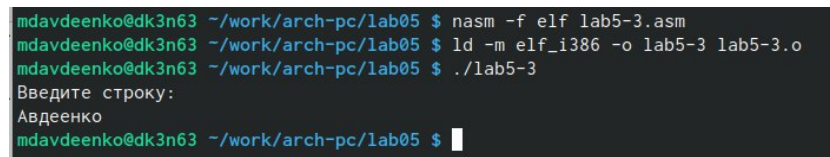
- вывести приглашение типа “Введите строку:”;
- ввести строку с клавиатуры (рис. 5.1).



```
GNU nano 6.4 /afs/.dk.sci.pfu.edu.ru/home/m/d/mdavdeenko/work/arch-pc/lab05/lab5-3.asm
;
; 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'
```

Рис. 5.1: Файл lab5-3.asm

2) Получила исполняемый файл и проверила его работу. На приглашение ввести строку ввела свою фамилию (рис. 5.2).

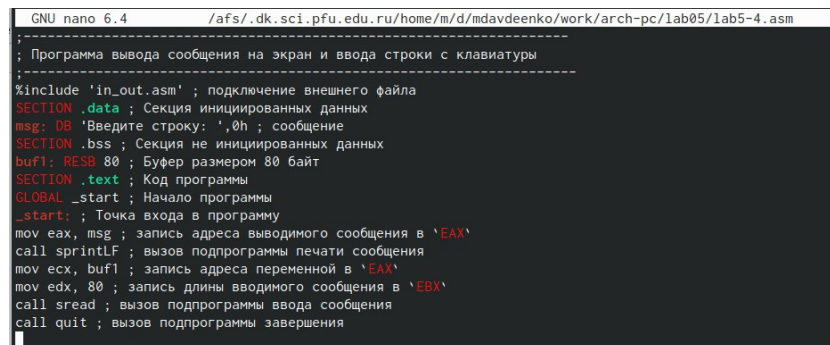


```
mdavdeenko@dk3n63 ~/work/arch-pc/lab05 $ nasm -f elf lab5-3.asm
mdavdeenko@dk3n63 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-3 lab5-3.o
mdavdeenko@dk3n63 ~/work/arch-pc/lab05 $ ./lab5-3
Введите строку:
Авдеенко
mdavdeenko@dk3n63 ~/work/arch-pc/lab05 $
```

Рис. 5.2: Файл lab5-3.asm с выводом фамилии

3) Создайте копию файла lab5-2.asm. Исправьте текст программы с использование под-программ из внешнего файла in\_out.asm, так чтобы она работала по следующему алгоритму:

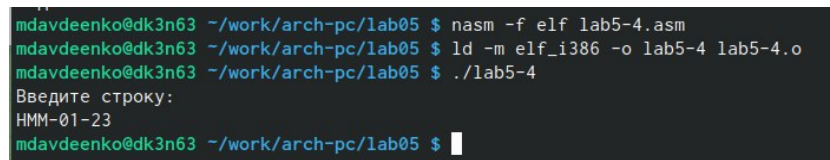
- вывести приглашение типа “Введите строку:”;
- ввести строку с клавиатуры (рис. 5.3).



```
GNU nano 6.4 /afs/.dk.sci.pfu.edu.ru/home/m/dmdavdeenko/work/arch-pc/lab05/lab5-4.asm
;-----
; 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 sprintf ; вызов подпрограммы печати сообщения
mov ecx, buf1 ; запись адреса переменной в 'ECX'
mov edx, 80 ; запись длины вводимого сообщения в 'EDX'
call sread ; вызов подпрограммы ввода сообщения
call quit ; вызов подпрограммы завершения
```

Рис. 5.3: Файл lab5-4.asm

4) Получите исполняемый файл и проверьте его работу. На приглашение ввести строку введите номер своей группы (рис. 5.4).



```
mdavdeenko@dk3n63 ~/work/arch-pc/lab05 $ nasm -f elf lab5-4.asm
mdavdeenko@dk3n63 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-4 lab5-4.o
mdavdeenko@dk3n63 ~/work/arch-pc/lab05 $ ./lab5-4
Введите строку:
HMM-01-23
mdavdeenko@dk3n63 ~/work/arch-pc/lab05 $
```

Рис. 5.4: Файл lab5-4.asm с выводом номера группы

## 6 Выводы

В ходе данной лабораторной работы были приобретены практические навыки работы в Midnight Commander.

## **Список литературы**