

# **Отчет по выполнению лабораторной работы №5**

**Дисциплина: архитектура компьютера**

Ефремова Полина Александровна

# 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>9</b>
<b>5</b>	<b>Задание для самостоятельной работы</b>	<b>18</b>
<b>6</b>	<b>Выводы</b>	<b>24</b>
	<b>Список литературы</b>	<b>25</b>

# Список иллюстраций

4.1	Midnight Commander . . . . .	9
4.2	Создание нового каталога . . . . .	10
4.3	Создание нового файла . . . . .	10
4.4	Открытие файла . . . . .	11
4.5	Редактирование файла . . . . .	12
4.6	Реализация программы . . . . .	13
4.7	Копирование . . . . .	14
4.8	Перенос . . . . .	14
4.9	Переименование . . . . .	15
4.10	Ввод программы в файл lab5-2 . . . . .	15
4.11	Создание и проверка программы . . . . .	16
4.12	Замена подпрограммы в файле . . . . .	16
4.13	Просмотр измененной команды . . . . .	17
5.1	Копия lab5-1 . . . . .	18
5.2	Изменение программы 1 . . . . .	19
5.3	Проверка 1 . . . . .	20
5.4	Копия lab5-2 . . . . .	21
5.5	Изменение программы 2 . . . . .	22
5.6	Проверка 2 . . . . .	23

## **Список таблиц**

# 1 Цель работы

Цель данной работы - это приобретение практических навыков работы в Midnight Commander, а также освоение инструкций языка ассемблера `mov` и `int`.

## 2 Задание

1. Изучение основ работы с Midnight Commander.
2. Продолжение знакомства со структурой программы на языке ассемблера NASM.
3. Выполнение заданий для самостоятельной работы с целью закрепления материала.

## 3 Теоретическое введение

**Midnight Commander** - популярный двухпанельный файловый менеджер, широко распространенный в UNIX-like операционных системах, он должен быть знаком каждому, кто хоть раз работал в консоли. Но, как показывает практика, не все администраторы в полной мере используют все возможности данного приложения, ограничиваясь только базовыми, что может приводить к определенным неудобствам. Поэтому сегодня мы расскажем о возможностях Midnight Commander и приемах, способных сделать работу в нем удобной и эффективной.

Основы навигации должны быть известны всем, но тем не менее повторимся: для перемещения внутри панелей следует использовать клавиши со стрелками, PageUp и PageDown прокручивают содержимое на размер экрана вверх или вниз, а Home и End перемещают в начало или конец списка, для переключения панелей используется Tab, сочетание клавиш Ctrl + o сворачивает панели и дает доступ к консоли, повторное нажатие этих же клавиш восстанавливает панели.

Нижнее меню предлагает ряд часто используемых действий, привязанных к функциональным клавишам Fn, цифра, написанная около пункта меню, обозначает номер функциональной клавиши, например, F5 - копирование, F8 - удаление. Клавиша F9 перемещает нас в верхнее меню менеджера, навигация по которому осуществляется с помощью клавиш со стрелками.

Разделы Левая панель и Правая панель верхнего меню идентичны и содержат настройки отображения панелей, также отсюда можно инициализировать внешнее FTP или SSH соединение, после чего панель будет отображать содержимое удаленного узла.

Меню Файл содержит основные действия над файлами, здесь вы можете выполнить базовые действия над файлами, а также быстро изменить права, владельца, создать жесткую или символическую ссылку. Обратите внимание, около команд практически везде стоят сочетания клавиш, рекомендуем их запомнить, что позволит работать гораздо быстрее, не прибегая каждый раз к навигации по меню.



## 4 Выполнение лабораторной работы

1. Скачиваю и открываю Midnight Commander (рис. 4.1).

```
paefremova@fedora:~$ mc
bash: mc: команда не найдена...
Установить пакет «mc», предоставляющий команду «mc»? [N/y] y

* Ожидание в очереди...
Следующие пакеты должны быть установлены:
gpm-libs-1.20.7-46.fc40.x86_64 Dynamic library for gpm
mc-1:4.8.31-1.fc40.x86_64 User-friendly text console file manager and visual shell
Продолжить с этими изменениями? [N/y] y

* Ожидание в очереди...
* Ожидание аутентификации...
* Ожидание в очереди...
* Загрузка пакетов...
* Запрос данных...
* Проверка изменений...
* Установка пакетов...
```

Рис. 4.1: Midnight Commander

2. Перехожу в каталог arch-rc, созданный при выполнении прошлой лабораторной работы. Создаю каталог lab05. (рис. 4.2).

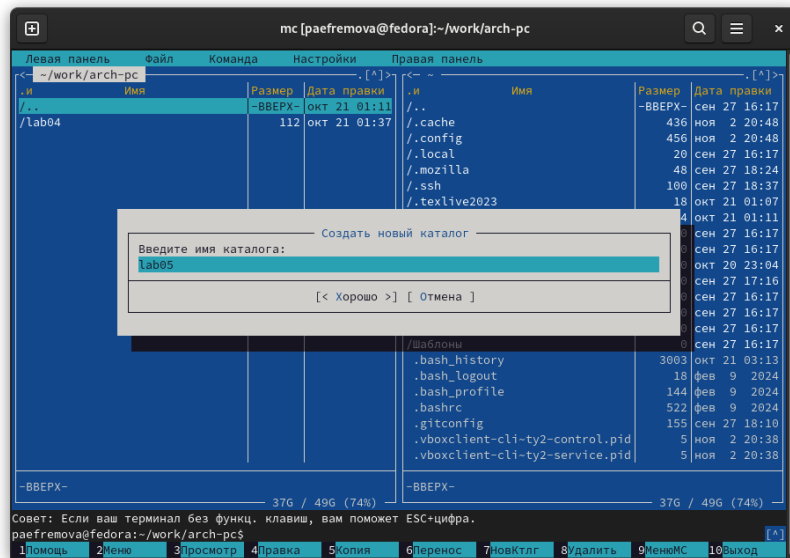


Рис. 4.2: Создание нового каталога

3. С помощью команды `touch` создаю новый файл. (рис. 4.3).

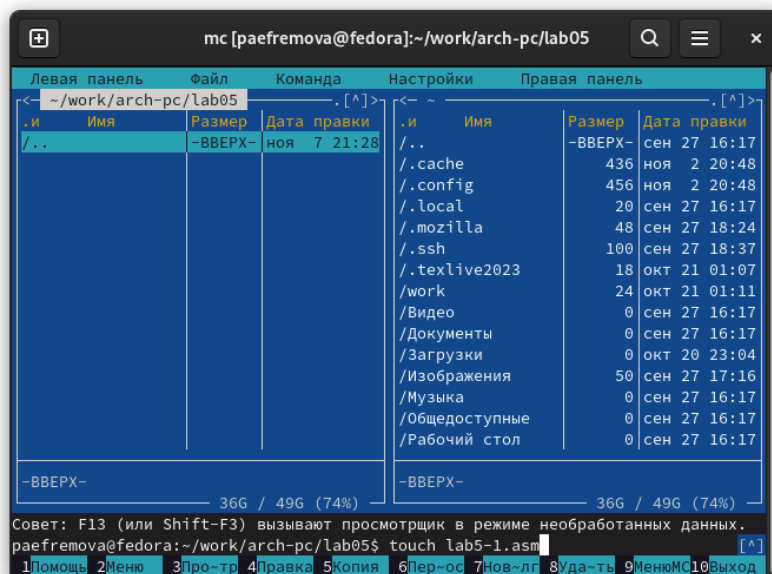


Рис. 4.3: Создание нового файла

4. Открываю файл (рис. 4.4).

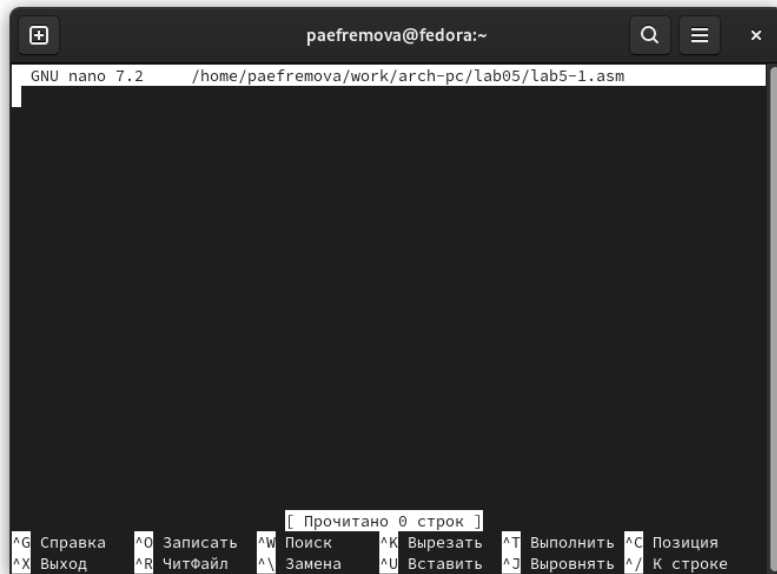
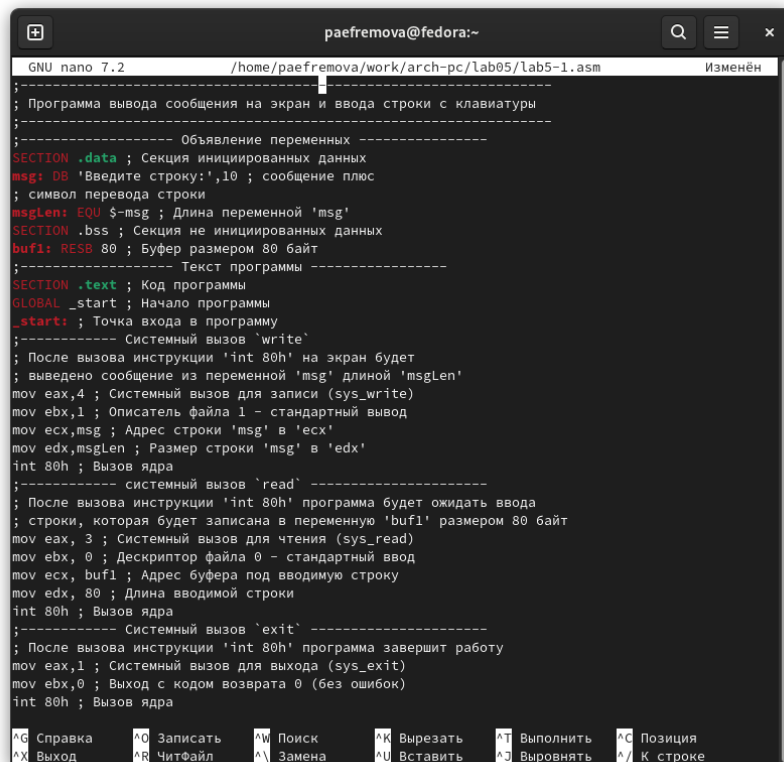


Рис. 4.4: Открытие файла

5. Ввожу текст программы из листинга 5.1, сохраняю изменения и закрываю файл. (рис. 4.5).

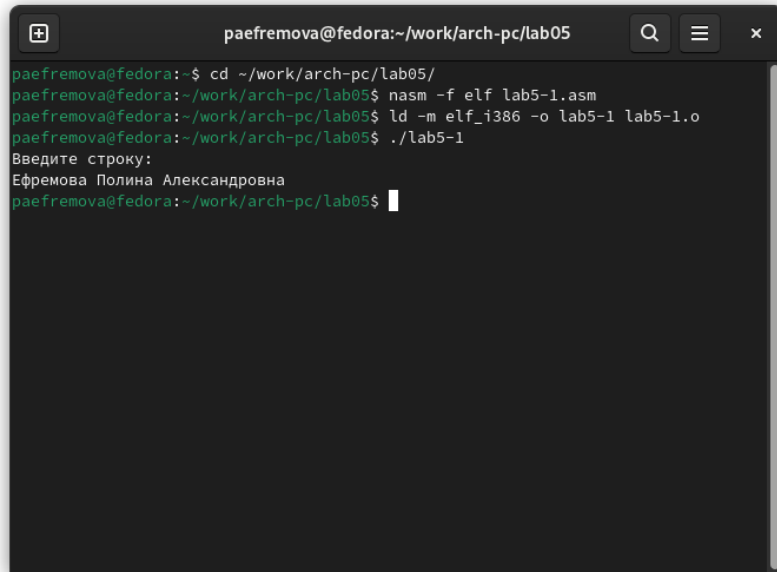


```
GNU nano 7.2 /home/paefremova/work/arch-pc/lab05/lab5-1.asm
; Программа вывода сообщения на экран и ввода строки с клавиатуры
;----- Объявление переменных -----
SECTION .data ; Секция инициализированных данных
msg: DB 'Введите строку:',10 ; сообщение плюс
; символ перевода строки
msgLen: EQU $-msg ; 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'
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

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

Рис. 4.5: Редактирование файла

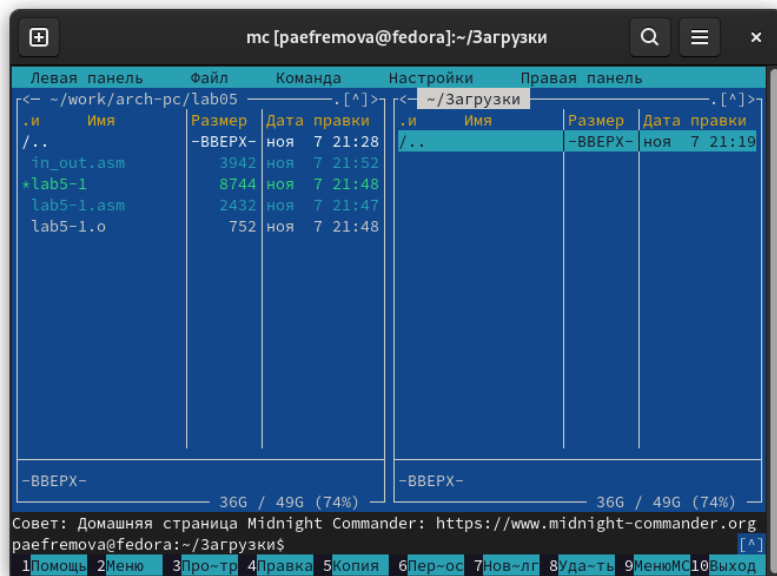
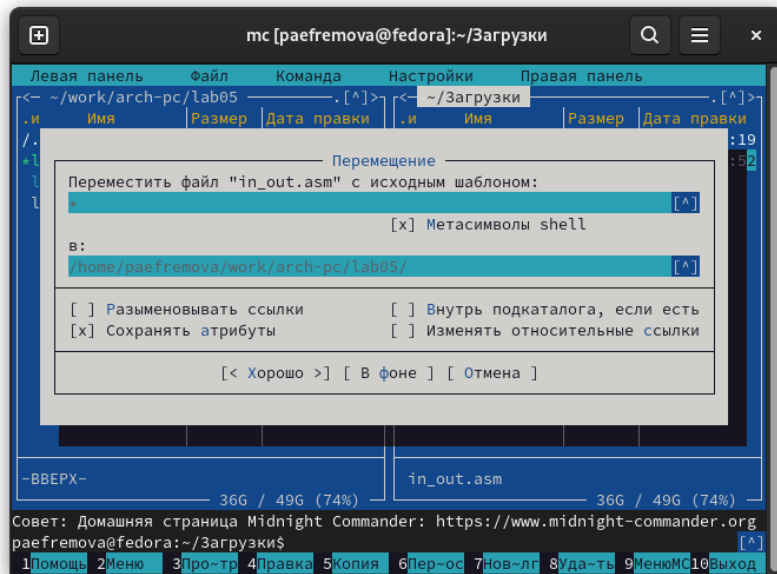
6. Оттранслирую текст программы lab5-1.asm в объектный файл. Выполняю компоновку объектного файла и запускаю получившийся исполняемый файл. (рис. 4.6).

A terminal window titled 'paefremova@fedora:~/work/arch-pc/lab05'. The terminal shows the following commands and output:

```
paefremova@fedora:~$ cd ~/work/arch-pc/lab05/
paefremova@fedora:~/work/arch-pc/lab05$ nasm -f elf lab5-1.asm
paefremova@fedora:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-1 lab5-1.o
paefremova@fedora:~/work/arch-pc/lab05$ ./lab5-1
Введите строку:
Ефремова Полина Александровна
paefremova@fedora:~/work/arch-pc/lab05$
```

Рис. 4.6: Реализация программы

7. С помощью функциональной клавиши F6 создаю копию файла lab5-1.asm с именем lab5-2.asm. Выделяю файл lab5-1.asm, Нажимаю клавишу F6 , ввожу имя файла lab5-2.asm и нажимаю клавишу Enter. (рис. 4.7). (рис. 4.8). (рис. 4.9).



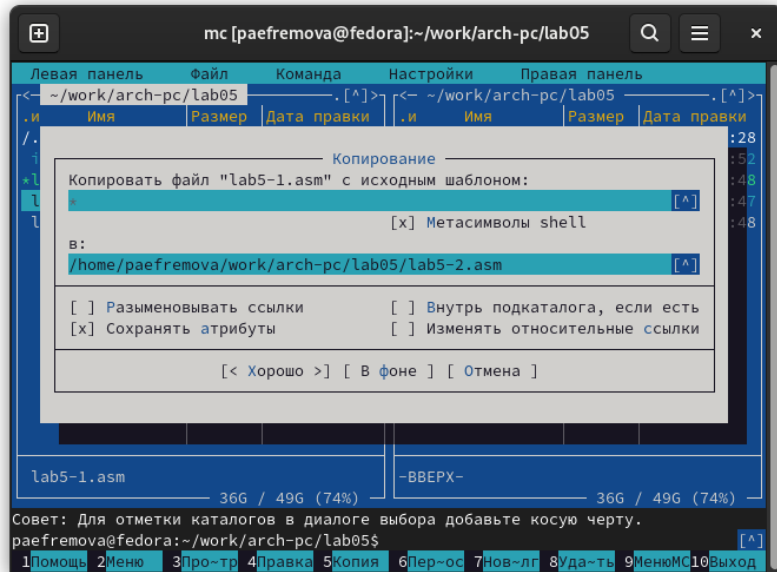


Рис. 4.9: Переименование

8.Исправляю текст программы в файле lab5-2.asm с использованием подпрограмм из внешнего файла in\_out.asm в соответствии с листингом 5.2. (рис. 4.10).

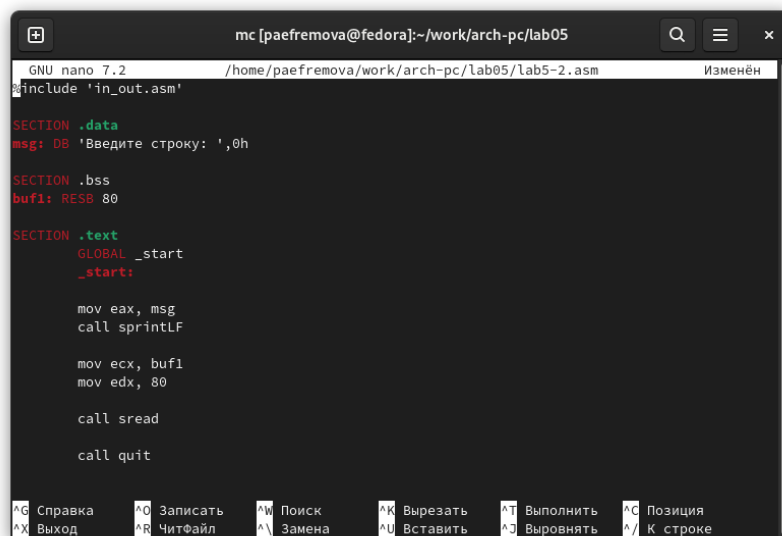
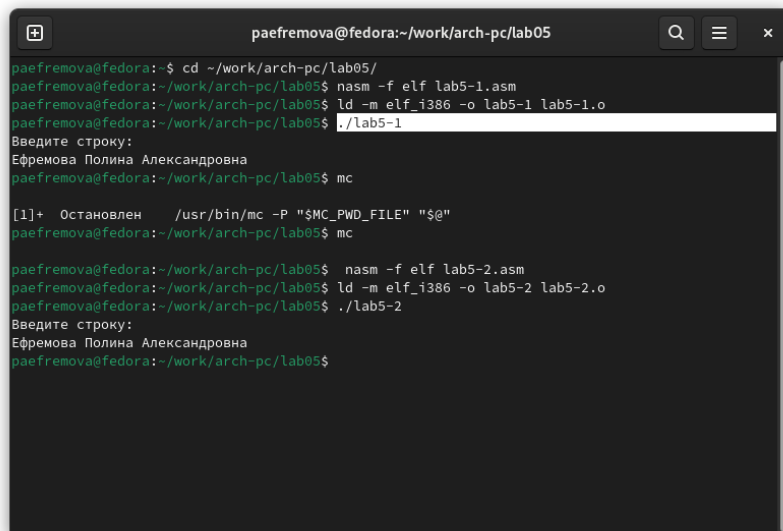


Рис. 4.10: Ввод программы в файл lab5-2

9.Создаю исполняемый файл и проверяю его работу. (рис. 4.11).



```
paefremova@fedora:~/work/arch-pc/lab05
paefremova@fedora:~/work/arch-pc/lab05$ cd ~/work/arch-pc/lab05/
paefremova@fedora:~/work/arch-pc/lab05$ nasm -f elf lab5-1.asm
paefremova@fedora:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-1 lab5-1.o
paefremova@fedora:~/work/arch-pc/lab05$ ./lab5-1
Введите строку:
Ефремова Полина Александровна
paefremova@fedora:~/work/arch-pc/lab05$ mc
[1]+  Остановлен /usr/bin/mc -P "$MC_PWD_FILE" "$@"
paefremova@fedora:~/work/arch-pc/lab05$ mc
paefremova@fedora:~/work/arch-pc/lab05$ nasm -f elf lab5-2.asm
paefremova@fedora:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-2 lab5-2.o
paefremova@fedora:~/work/arch-pc/lab05$ ./lab5-2
Введите строку:
Ефремова Полина Александровна
paefremova@fedora:~/work/arch-pc/lab05$
```

Рис. 4.11: Создание и проверка программы

10. Заменяв в файле подпрограмму `sprintLF` на `sprint` (рис. 4.12).



```
GNU nano 7.2 /home/paefremova/work/arch-pc/lab05/lab5-2.asm
#include 'in_out.asm'

SECTION .data
msg: DB 'Введите строку: ',0h

SECTION .bss
buf1: RESB 80

SECTION .text
GLOBAL _start
_start:

    mov eax, msg
    call sprint

    mov ecx, buf1
    mov edx, 80

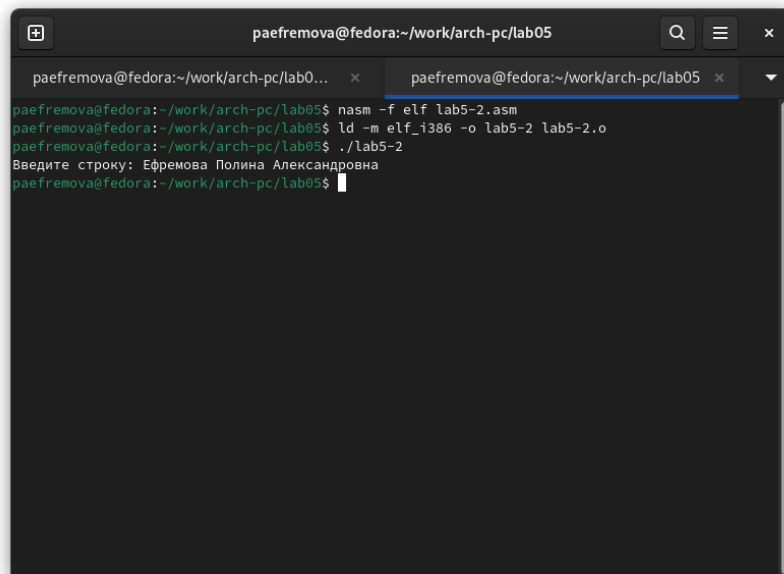
    call sread

    call quit
```

Рис. 4.12: Замена подпрограммы в файле

11.Понимаю, что теперь текст вводится в той же строке, что и запрос команды.  
(рис. 4.13).





A terminal window with a dark background and light green text. The window title is 'paefremova@fedora:~/work/arch-pc/lab05'. The terminal shows the following commands and output:

```
paefremova@fedora:~/work/arch-pc/lab05$ nasm -f elf lab5-2.asm
paefremova@fedora:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-2 lab5-2.o
paefremova@fedora:~/work/arch-pc/lab05$ ./lab5-2
Введите строку: Ефремова Полина Александровна
paefremova@fedora:~/work/arch-pc/lab05$
```

Рис. 4.13: Просмотр измененной команды

## 5 Задание для самостоятельной работы

1. Создаю копию файла lab5-1.asm. (рис. 5.1).

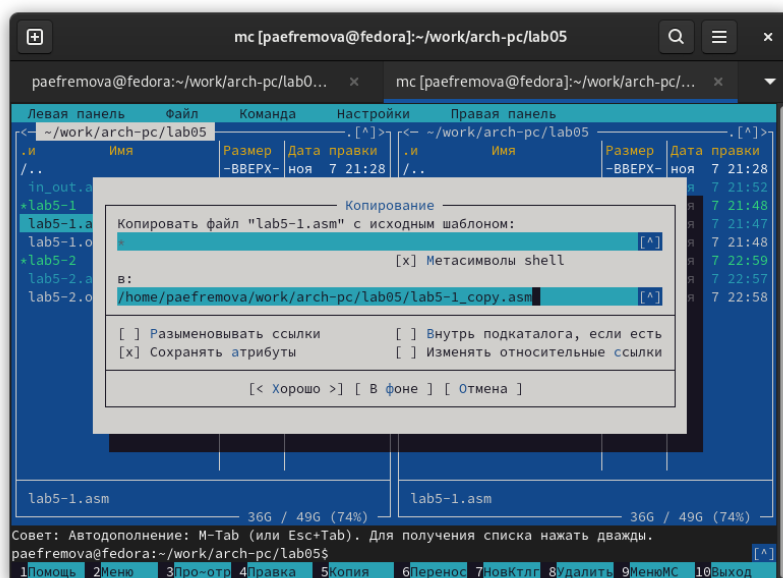
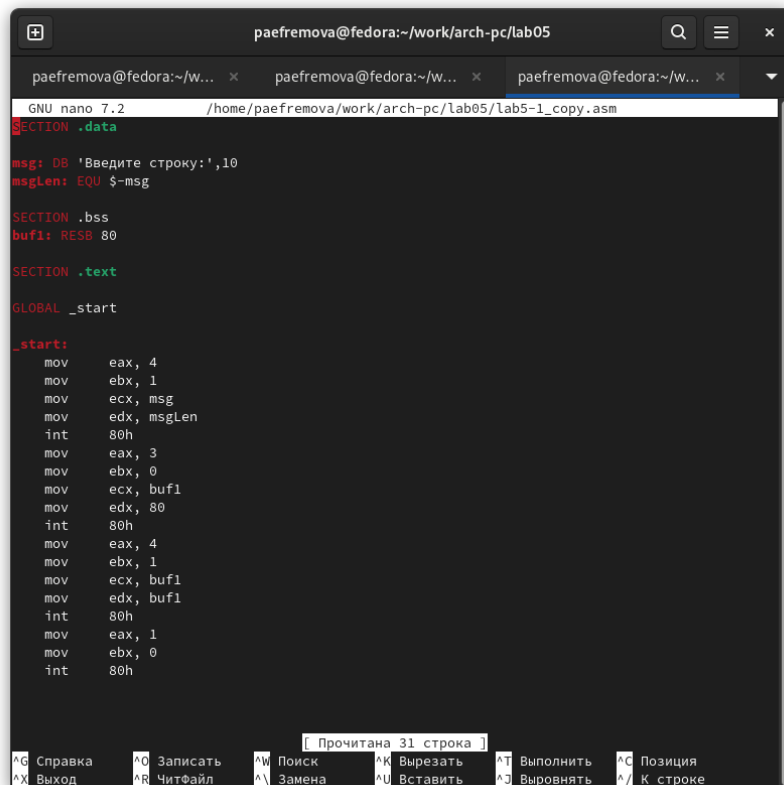


Рис. 5.1: Копия lab5-1

2. Вношу 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; • вывести введенную строку на экран. (рис. 5.2).



```
GNU nano 7.2 /home/paefremova/work/arch-pc/lab05/lab5-1_copy.asm
SECTION .data
msg: DB 'Введите строку:',10
msgLen: EQU $-msg

SECTION .bss
buf1: RESB 80

SECTION .text

GLOBAL _start

_start:
    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. 5.2: Изменение программы 1

Ниже показан код программы:

SECTION .data

msg: DB 'Введите строку:',10 msgLen: EQU \$-msg

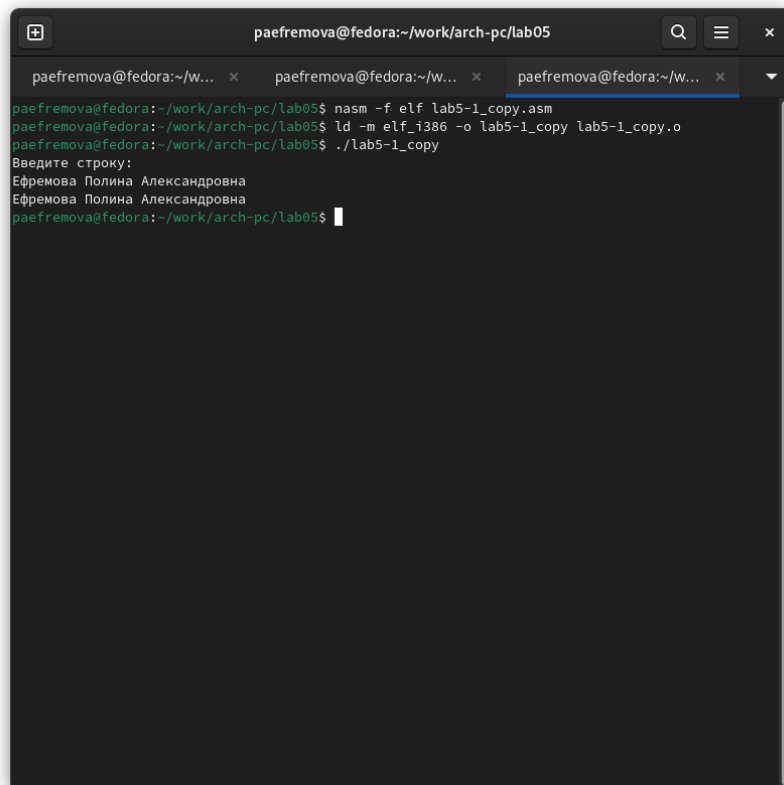
SECTION .bss buf1: RESB 80

SECTION .text

GLOBAL \_start

\_start: 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. Получаю исполняемый файл и проверяю его работу. (рис. 5.3).



A terminal window titled 'paefremova@fedora:~/work/arch-pc/lab05'. The window contains the following text:

```
paefremova@fedora:~/work/arch-pc/lab05$ nasm -f elf lab5-1_copy.asm
paefremova@fedora:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-1_copy lab5-1_copy.o
paefremova@fedora:~/work/arch-pc/lab05$ ./lab5-1_copy
Введите строку:
Ефремова Полина Александровна
Ефремова Полина Александровна
paefremova@fedora:~/work/arch-pc/lab05$
```

Рис. 5.3: Проверка 1

4. Создаю копию файла lab5-2.asm. (рис. 5.4).

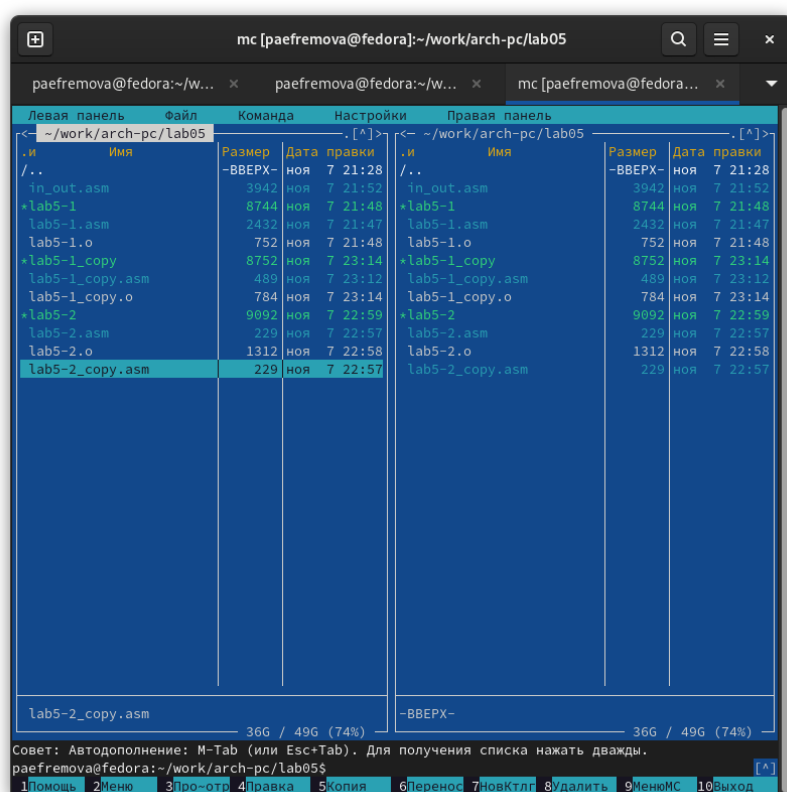
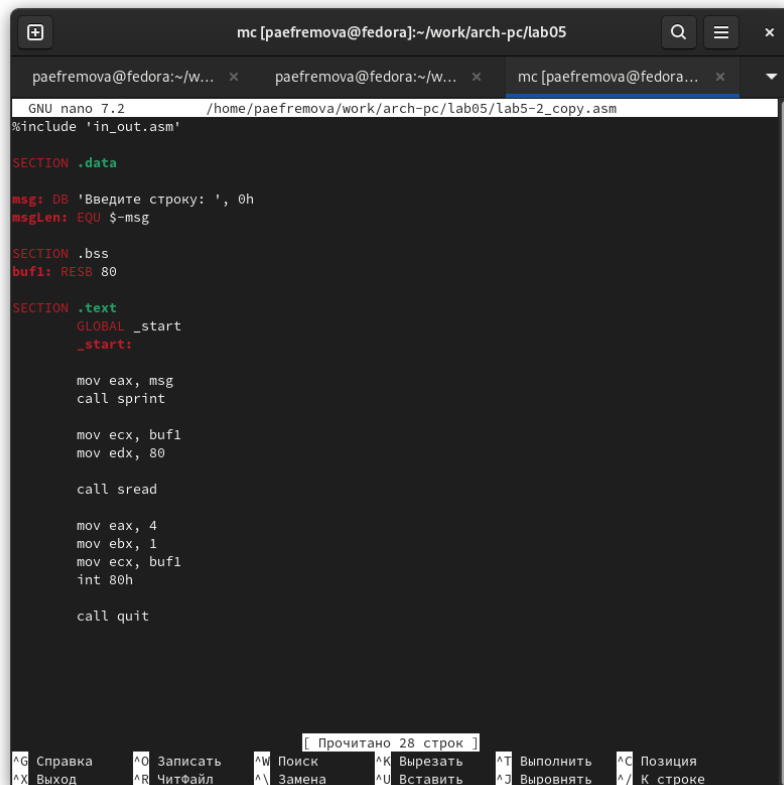


Рис. 5.4: Копия lab5-2

- Исправляю текст программы с использованием подпрограмм из внешнего файла `in_out.asm`, так чтобы она работала по следующему алгоритму:
  - вывести приглашение типа “Введите строку:”;
  - ввести строку с клавиатуры;
  - вывести введённую строку на экран. (рис. 5.5).



```
GNU nano 7.2 /home/paefremova/work/arch-pc/lab05/lab5-2_copy.asm
#include 'in_out.asm'

SECTION .data
msg: DB 'Введите строку: ', 0h
msgLen: EQU $-msg

SECTION .bss
buf1: RESB 80

SECTION .text
GLOBAL _start
_start:

    mov eax, msg
    call sprint

    mov ecx, buf1
    mov edx, 80

    call sread

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

    call quit
```

Прочитано 28 строк

AG Справка    ^O Записать    ^W Поиск    ^K Вырезать    ^T Выполнить    ^G Позиция  
AX Выход       ^R ЧитФайл    ^N Замена    ^U Вставить    ^D Выровнять    ^V К строке

Рис. 5.5: Изменение программы 2

Ниже показан код программы:

```
%include 'in_out.asm'
```

```
SECTION .data
```

```
msg: DB 'Введите строку:', 0h msgLen: EQU $-msg
```

```
SECTION .bss buf1: RESB 80
```

```
SECTION .text GLOBAL _start _start:
```

```
mov eax, msg
```

```
call sprint
```

```
mov ecx, buf1
```

```
mov edx, 80
```

```
call sread
```

```
mov eax, 4
```

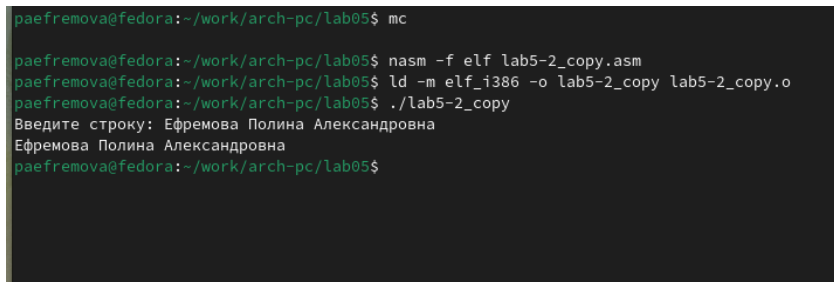
```
mov ebx, 1
```

```
mov ecx, buf1
```

```
int 80h
```

```
call quit
```

6. Получаю исполняемый файл и проверяю его работу. (рис. 5.6).



```
paefremova@fedora:~/work/arch-pc/lab05$ mc
paefremova@fedora:~/work/arch-pc/lab05$ nasm -f elf lab5-2_copy.asm
paefremova@fedora:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-2_copy lab5-2_copy.o
paefremova@fedora:~/work/arch-pc/lab05$ ./lab5-2_copy
Введите строку: Ефремова Полина Александровна
Ефремова Полина Александровна
paefremova@fedora:~/work/arch-pc/lab05$
```

Рис. 5.6: Проверка 2

## 6 Выводы

При выполнении данной лабораторной работы я получила большое количество теоретической информации относительно Midnight Commander, которую закрепила после, выполняя базовые команды. Кроме этого, я также освоила некоторые инструкции языка ассемблера.



## Список литературы

1. Лабораторная работа №5
2. Осваиваем эффективную работу в Midnight Commander - Записки IT специалиста